Multitask learning and data distribution search in visual relationship recognition

by

Shane Josias

Thesis presented in partial fulfilment of the requirements for the degree of Master of Science (Applied Mathematics) in the Faculty of Science at Stellenbosch University

Supervisor: Prof. Willie Brink

March 2020

Stellenbosch University https://scholar.sun.ac.za

Declaration

By submitting this thesis electronically, I declare that the entirety of the work contained therein is my own, original work, that I am the sole author thereof (save to the extent explicitly otherwise stated), that reproduction and publication thereof by Stellenbosch University will not infringe any third party rights and that I have not previously in its entirety or in part submitted it for obtaining any qualification.

Date: March 2020 .....

Copyright © 2020 Stellenbosch University
All rights reserved.

Stellenbosch University https://scholar.sun.ac.za

Abstract

An image can be described by the objects within it, as well as the interactions between those objects. A pair of object labels together with an interaction label can be assembled into what is known as a visual relationship, represented as a triplet of the form (subject, predicate, object). Recognising visual relationships in a given image is a challenging task, owing to the combinatorially large number of possible relationship triplets which lead to a so-called extreme classification problem, as well as a very long tail found typically in the distribution of those possible triplets.

We investigate the efficacy of four strategies that could potentially address these issues. Firstly, instead of predicting the full triplet we opt to predict each element separately. Secondly, we investigate the use of shared network parameters to perform these separate predictions in a basic multitask setting. Thirdly, we extend the multitask setting by including an online ranking loss that acts on a trio of samples (an anchor, a positive sample, and a negative sample). Semi-hard negative mining is used to select negative samples. Finally, we consider a class-selective batch construction strategy to expose the network to more of the many rare classes during mini-batch training. We view semi-hard negative mining and class-selective batch construction as training data distribution search, in the sense that they both attempt to carefully select training samples in order to improve model performance. In addition to the aforementioned strategies, we also introduce a means of evaluating model behaviour in visual relationship recognition. This evaluation motivates the use of semantics.

Our experiments demonstrate that batch construction can improve performance on the long tail, possibly at the expense of accuracy on the small number of dominating classes. We also find that a basic multitask model neither improves nor impedes performance in any significant way, but that its smaller size may be beneficial. Moreover, multitask models trained with a ranking loss yield a decrease in performance, possibly due to limited batch sizes.

Stellenbosch University https://scholar.sun.ac.za

Opsomming

'n Beeld kan beskryf word deur die voorwerpe daarin, asook die interaksies tussen daardie voorwerpe. Twee voorwerpetikette saammet 'n interaksie-etiket staan bekend as 'n visuele verwantskap, en word voorgestel met 'n drieling van die vorm (onderwerp, predikaat, voorwerp). Die herkenning van visuele verwantskappe in 'n gegewe beeld is 'n uitdagende taak, te danke aan die kombinatoriese groot aantal moontlike verwantskap-drielinge, wat lei tot 'n sogenaamde ekstreme klassifikasieprobleem, sowel as 'n baie lang stert wat tipies in die verspreiding van daardie moontlike drielinge voorkom.

Ons ondersoek die doeltreffendheid van vier strategieë om hierdie probleme aan te pak. Eerstens, in plaas daarvan om die volledige drieling te voorspel, kies ons om elke element afsonderlik te voorspel. Tweedens ondersoek ons die gebruik van gedeelde netwerkparameters om hierdie afsonderlike voorspellings in 'n basiese multitaak-opstelling uit te voer. Derdens brei ons die multitaak-opstelling uit deur 'n aanlyn rang-verliesfunksie in te sluit, gedefinieër op 'n trio van datapunte ('n anker, 'n positiewe voorbeeld en 'n negatiewe voorbeeld). Semi-moeilike negatiewe ontginning word gebruik om negatiewe voorbeelde te selekteer. Laastens word daar gekyk na 'n klas-selektiewe bondelkonstruksie-strategie om die netwerk bloot te stel aan meer van die seldsame klasse tydens mini-bondel afrigting. Ons beskou semi-moeilike negatiewe ontginning en klas-selektiewe bondelkonstruksie as vorme van 'n dataverspreidings-soektog. Albei poog om afrig-datapunte noukeurig te kies om die model se prestasie te verbeter. Benewens die bogenoemde strategieë, stel ons ook 'n manier voor om modelgedrag in die herkenning van visuele verwantskappe te evalueer. Hierdie evaluering motiveer die gebruik van semantiek.

Ons eksperimente demonstreer dat bondelkonstruksie prestasie op die lang stert kan verbeter, moontlik ten koste van akkuraatheid op die klein aantal dominante klasse. Ons vind ook dat 'n basiese multitaakmodel nie die prestasie op 'n beduidende manier verbeter of belemmer nie, maar dat die kleiner modelgrootte daarvan voordelig kan wees. Boonop lei multitaakmodelle wat met 'n rang-verliesfunksie afgerig word, tot 'n laer prestasie, moontlik as gevolg van beperkte bondelgroottes.

Stellenbosch University https://scholar.sun.ac.za

Acknowledgements

I would like to express my sincere appreciation to the following individuals and institutions who contributed to the completion of this research:

- my supervisor, Prof. Willie Brink, whose guidance and attention to detail has played a vital role in the culmination of this thesis;
- the CSIR/SU Centre for Artificial Intelligence Research (CAIR) for financial support;
- Peter Thompson, Bronwyn Dumbleton, Burger Becker, Herman Kamper, Shaun Wurdeman, Christiaan Landman and Joe Stoker for invaluable discussions, advice, and encouragement;
- Piotr Bialecki who was active on PyTorch forums and Adam Bielski whose open-sourced code provided the basis for the trio sampling used in this thesis.

Stellenbosch University https://scholar.sun.ac.za

Contents

Stellenbosch University https://scholar.sun.ac.za

Stellenbosch University https://scholar.sun.ac.za

Chapter 1

Introduction

There exists a variety of effective methods for locating and classifying objects in an image [1; 2], which could form part of an image understanding pipeline. To further develop such a pipeline we might want to consider methods for recognising the interaction or relationship between different objects in the same image.

A visual relationship is defined as a triplet of the form (subject, predicate, object) that describes some visible interaction between a pair of objects in an image. The image in Figure 1.1, for example, contains the visual relationship (`boy`, `on top of`, `surfboard`). Such visual relationships can be used to construct a scene graph representation of an image [3], for further visual reasoning in tasks such as image retrieval, visual question answering, and automated surveillance.

Visual relationship recognition is the problem of producing (subject, predicate, object) triplets for a given image. It is often coupled with object localisation, but the focus of this thesis is on the classification task and we will therefore assume knowledge of tight bounding boxes around pairs of objects, as illustrated in Figure 1.1. Bounding boxes around objects can be generated by an off-the-shelf object detector (e.g. [1]) and merged pairwise in a straightforward manner.

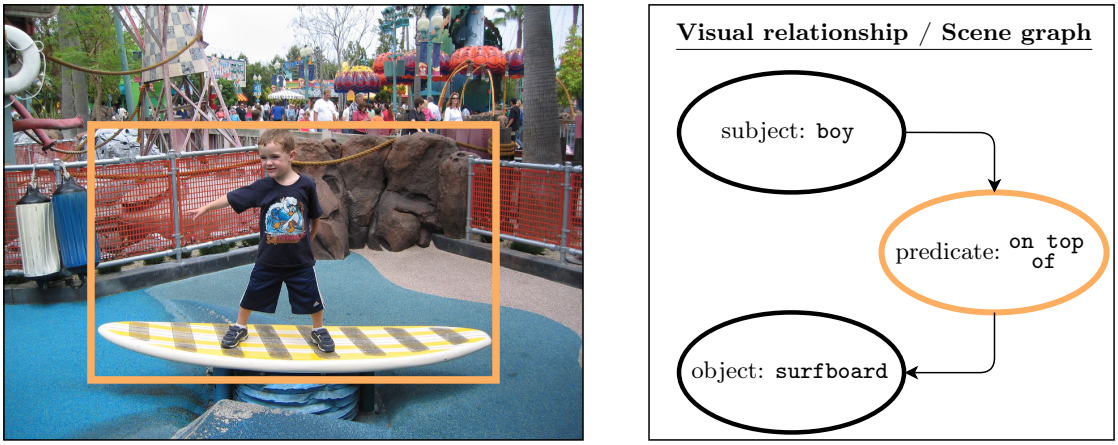


Figure 1.1: An example of visual relationship recognition. The task is to label the subject, the predicate (relationship) and the object, given an image and a bounding box around a pair of objects. The visual relationship (`boy`, `on top of`, `surfboard`) can then be used to construct a scene graph representation of an image.

Visual relationship recognition is challenging for a number of reasons. Firstly, the number of possible relationships explodes combinatorially and leads to what is known as an

Stellenbosch University https://scholar.sun.ac.za

extreme multiclass classification problem. For example, 100 possible subject and object labels, and 70 possible predicate labels, amount to 700,000 possible triplets. A high number of classes complicates the classification problem for the following reasons.

1. It is near impossible to collect data that represents all classes. In fact, one of the first datasets on visual relationship recognition, called VRD [4], represents a 700,000 class problem (taking all possible combinations of subjects, predicates and objects into account) but contains only around 15,000 unique visual relationships. A substantially larger dataset called Visual Genome [5] contains 75,729 unique objects but only 40,480 unique visual relationships.

2. Neural networks are commonly used to solve classification problems, and apply the softmax function over its output. A neural network with a 700,000-dimensional layer output contains far too many weights and makes training computationally infeasible. In other domains, such as word embeddings for computational linguistics, techniques like hierarchical softmax and negative mining have been developed to overcome this.

The second challenge is that the distribution of visual relationships in a dataset typically exhibits a very long tail: the vast majority of possible triplets might occur only a few times (or never) in the training set, while a small number might be frequent. An example of such a distribution is shown in Figure 1.2. This behaviour is likely due to the fact that the distribution of individual elements of the relationship triplet also exhibit a long tail. A long tail is problematic for optimisation based learning, because an undesired local optimum to the objective can be found quickly by merely predicting the dominant classes most often.

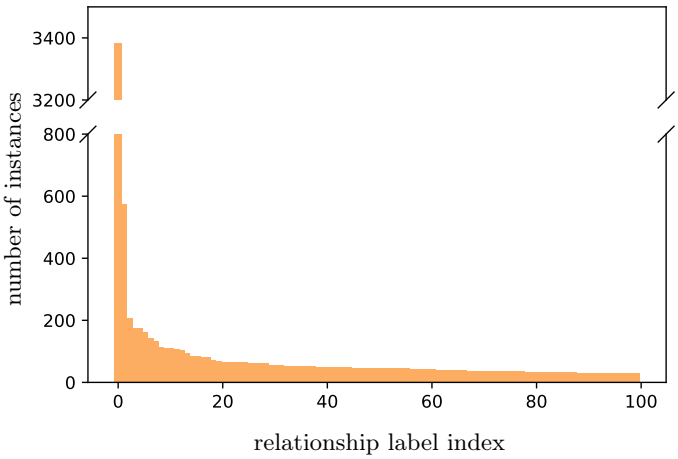


Figure 1.2: Ordered histogram of the relationship instances containing each subject label, predicate label and object label, across the VRD dataset [4]. Only the top 100 (out of 15,000) relationships are shown, and already the long tail is apparent.

The third challenge in visual relationship recognition is that predicates tend to be somewhat more abstract than the subjects and objects, making their visual representations more difficult to model and recognise.

Stellenbosch University https://scholar.sun.ac.za

Another challenge is that there seems to be an inherent ambiguity in the labelling of visual relationships. For example, the visual relationship (`boy`, `on top of`, `surfboard`) could legitimately be labelled as (`boy`, `riding`, `surfboard`) or (`surfboard`, `under`, `boy`). This is further discussed, with examples, in Chapters 3 and 4. Having multiple semantically correct classifications of a visual relationship, while typically having only a single ground truth label in the dataset, makes both modelling and evaluating visual relationship recognition difficult. It is furthermore not clear whether there is enough information in image data alone to tackle the problem.

1.1 Aims and contributions

We investigate a number of strategies to deal with the issues above. To address the combinatorially large set of possible classes, we design models that separately predict the elements of a triplet, instead of a single prediction of the complete triplet.

This strategy allows for a multitask design where the different elements can be predicted with shared model parameters, potentially resulting in inductive transfer and data amplification [6] for improved generalisation. Multitask learning is viewed as a type of transfer learning, where knowledge is transferred across tasks.

For model training we also implement class-selective mini-batch construction through a type of training data distribution search, in an effort to better capture the long-tailed distribution over visual relationships. Training data distribution search attempts to find ways to better select training samples so that models can generalise as best as possible.

We compare the performance of our class-selective batch construction strategy against standard uniformly random batch sampling, and also our multitask model against multiple single-task models. The multitask setup is extended to include the use of a ranking loss function, which learns a visual embedding space using a similarity measure. The ranking loss is minimised in an online fashion and also makes use of a type of training data distribution search, where at training time examples are carefully selected to improve learning. An embedding space allows for an ability to perform few- and zero-shot learning. Since visual relationships contain many classes with few and no examples, such a paradigm seems useful.

All methods explored in this thesis make an implicit assumption that there is enough information in the image data to deal with the semantic ambiguity inherent in visual relationships, but we present arguments for the inclusion of a language model to deal with this issue.

The contributions of this work can be summarised as follows.

1. We show that batch construction is useful as a simple strategy for improved performance on underrepresented relationships (the long tail of the distribution).

2. We demonstrate that multitask learning is effective at reducing model complexity, without a significant positive or negative impact on performance.

Stellenbosch University https://scholar.sun.ac.za

3. We show that the minimisation of an online ranking loss can lead to embeddings that give improved performance on underrepresented classes in relatively simple domains, but not so in visual relationship recognition.

4. Finally, we introduce a performance metric with the aim of better understanding model behaviour in visual relationship recognition.

Part of this work has been accepted at a peer-reviewed conference:

- S. Josias, W. Brink. Multitask learning and batch construction in visual relationship recognition. *SAUPEC/RobMech/PRASA Conference*, January 2020.

1.2 Related work

The literature on visual relationship recognition can be grouped broadly into three common approaches. The first involves the learning of a visual-semantic embedding space, through imposing criteria such as small distances between similar relationships [4], or modelling a relationship as vector translation between embedded objects [7], or by minimising a triplet-softmax loss [8]. A visual-semantic embedding allows for few- and zero-shot learning, and could therefore be suited for modelling a long-tailed distribution, but a separate classifier would still need to be trained on top of the embedding. Using language in the modelling process also attempts to deal with the inherent ambiguity. In some sense, language places a prior on which visual relationships should be classified as correct, and potentially minimises the semantic ambiguity.

The second common approach attempts to generate the scene graph, or collection of interconnected relationships, directly. Xu et al. [9] perform graph inference with a structural recurrent neural network and an iterative message passing scheme to refine its predictions. Zellers et al. [10] observe that natural images usually have certain kinds of structural regularities, which they dub "motifs", and propose stacked neural networks (MotifNets) to predict graph elements and an LSTM to encode global context. Further examples of this approach include the use of associative embeddings [11], graph parsing neural networks [12], and Graph R-CNN [13]. Woo et al. [14] improve on graph generation strategies by designing an explicit relational reasoning module. Generating a scene graph is more direct than the visual-semantic embedding approach, and end-to-end training to accomplish the intended task directly can lead to superior performance.

The third approach, and the one most relevant to this thesis, treats the prediction of each element of the relationship triplet as its own classification task. Some works use multi-stream architectures for each task [15; 16; 17; 18], while others employ a single multitask scheme [19; 20] similar to what we will investigate.

There seems to be a central theme of transferring knowledge for improved performance, through message passing, global context cues, or inductive transfer in multitask learning. The multi-stream and multitask settings can deal with the huge number of classes in visual relationship recognition, by making use of multiple outputs of smaller dimensions. It does remain unclear whether multitask learning could necessarily provide better

Stellenbosch University https://scholar.sun.ac.za

performance. Existing approaches also tend to build very large systems, with many parameters, and it is usually not clear exactly how the long tail of typical datasets are dealt with. We have not yet come across visual relationship recognition approaches that deal with data distribution searches during training.

A significant effort is also being made to construct richer datasets that allow for better learning of visual relationships. The first major dataset released is called VRD [4]. It contains 5,000 images with around 15,000 unique visual relationships, with predicates belonging to one of five categories: action, spatial, preposition, comparative and verb. A much larger dataset called Visual Genome [5] was later introduced by Krishna et al. and contains 108,077 images with 40,480 unique relationships. While the number of images in Visual Genome is greater than VRD, the total number of visual relationship classes have also increased. The long tail distribution seems to be inherent in the problem of visual relationship recognition, and can be exacerbated when collecting more data. The Google Open Images Challenge attempts to find a middle ground by considering 329 possible relationship triplets with 375,000 visual relationship instances [21].

1.3 Thesis overview

The remainder of this documents is arranged as follows.

Chapter 2 discusses a fundamental challenge that exists when modelling high-dimensional data such as natural images. The idea that representations of data is important in classification tasks is presented and the use of neural networks is motivated. Thereafter, the specific strategies for visual relationship recognition employed in this thesis are presented. The chapter concludes with implementation details for the models used.

Chapter 3 introduces the datasets that are used in the thesis, for a more concrete understanding of the challenges in visual relationship recognition. Thereafter selected performance metrics are discussed. It is important to consider metrics that will reflect performance on the underrepresented classes. This chapter also contains preliminary results on common image classification tasks, to motivate our design choices.

Chapter 4 presents our results on the task of visual relationship recognition. We evaluate and discuss the performance of the strategies developed in Chapter 2, using the metrics introduced in Chapter 3. Results shown are of both a quantitative and qualitative nature.

Chapter 5 offers concluding remarks and suggests paths that can be taken as future work.

Stellenbosch University https://scholar.sun.ac.za

Chapter 2

Model design

Images are high-dimensional, which makes it difficult to obtain a classifier that generalises well. One possible solution to this is to find lower-dimensional representations of image data, that a classifier can use in order to generalise more effectively. Bengio et al. [22] suggest that representations of data are what drives success in machine learning.

This chapter explores the reasons that make high-dimensional data, such as images, difficult to model. Neural networks are then introduced as a means of learning representations of data. We motivate the use of neural networks in image classification by highlighting the assumptions and properties of representations that are learned by those networks. We can view the problem of visual relationship recognition as finding representations of data such that a classifier can generalise over an extremely large number of classes from a long-tailed distribution. We then introduce the following strategies which are investigated as a means of finding representations of data:

1. neural networks trained with mini-batching;
2. neural networks trained with class-selective batch construction;
3. a basic multitask learning paradigm;
4. hierarchical multitask learning with an online ranking loss;
5. split-multitask learning with an online ranking loss.

For each of these strategies we provide background theory, motivation and implementation details. We note that items 2, 4 and 5 are examples of training data distribution search techniques.

2.1 Data representation and neural networks

2.1.1 Curse of dimensionality

Classical machine learning techniques would have us handcraft features for a task like classification. This requires expert domain knowledge, and may not explain all factors of variation. As a result, and since domain knowledge can be scarce or expensive, models are often not particularly suited to generalisation. An issue is that many forms of data, especially images, are high in dimension. An example input to many image classification neural networks is an image with dimension $224 \times 224 \times 3$, in height, width and number of channels respectively. Such an image, when flattened, is a vector of dimension 150,528.

Stellenbosch University https://scholar.sun.ac.za

As the number of dimensions increases, the number of possible instances of variables that span this space increases exponentially and generalising to new examples becomes exceedingly difficult. This is a statistical challenge [23]. The number of possible instances of an image is far greater than the number of training samples that would be available in a practical setting. In other words, training data tends to sparsely cover the space, so that an unseen test sample would be unlikely to lie in the vicinity of a training sample.

2.1.2 The manifold hypothesis

There is a hypothesis stating that high-dimensional data, like images, form lower-dimensional manifolds in some embedding space [24]. This is somewhat intuitive in the case of images. We can imagine a set of continuous transformations that would transform one image to another and form continuous curves in image space [25]. The transformations can take the form of a change in lighting, a shift in the location of an object in the image, or some other change in the pixel values. Moreover, the probability distribution over images is highly concentrated. If we were to randomly sample pixel values for images, the probability that it would resemble a natural (or sensible) image is close to zero. So, natural images that a machine learning model might expect to encounter tend to sparsely cover the entire image space.

We can assume that, by approximation, most of the image space consists of non-natural images, and that natural images occur only along a collection of lower-dimensional manifolds. Olah [25] suggests that the problem of classification can be reduced to disentangling these manifolds. In other words, it may be desirable to obtain appropriate or useful representations of the input data so as to improve downstream tasks like visual relationship recognition. Neural networks create representations by manipulating the data manifold (they are functions defined on the domain of the data manifold) and the way they do so depends on the architecture, loss function and training procedure.

2.1.3 Neural networks

In image classification the goal is to obtain a mapping from an input image to a category label. A neural network can be viewed as a learnable approximation to this mapping. For input data $x \in \mathbb{R}^q$, let $f_w : \mathbb{R}^q \to \mathbb{R}^d$, be a differentiable neural network parameterised by w. We use this notation to refer to a neural network throughout the chapter.

A standard feedforward network [23] (with fully-connected layers) is represented by an input layer, at least one hidden layer, and an output layer, as seen in Figure 2.1. A neuron in each layer is connected to every neuron in a subsequent layer by a distinct edge. Consider the neuron h in Figure 2.1. It takes a weighted sum of its inputs, adds a bias and passes it to every neuron in the following layer. This is a linear function, however. To introduce nonlinearity (because it is unlikely that the soughtafter mapping to lower-dimensional manifolds is linear), the output is first passed through a nonlinear activation function σ. Common activation functions include ReLU, sigmoid, and tanh. Each layer can be thought of as an operation defined on the inputs. In this way, every layer creates a representation of the input data, and the collection of layers thus creates a sort of hierarchical representation. We are interested in mapping the input data to

Stellenbosch University https://scholar.sun.ac.za

some categorical label in the classification task. When training a neural network, the weights and biases are updated through gradient based optimisation so that this mapping is sufficiently approximated.

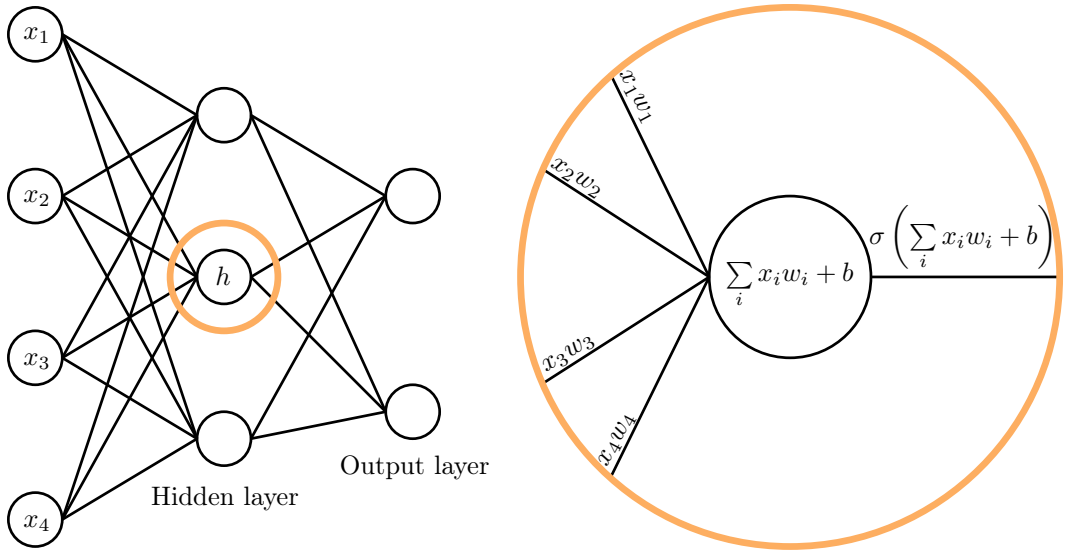


Figure 2.1: A fully-connected feedforward neural network. At each neuron, except for the input layer, a weighted sum of the inputs is taken and passed through an activation function before being sent to neurons in subsequent layers.

2.1.4 Convolutions and pooling

A special type of feedforward neural network is a convolutional neural network (CNN). Convolution can be described as an aggregation of the weighted sum of image pixels with a filter that is smaller than the image. For each pixel in the output of a particular layer, the kernel is centred at a particular pixel in the input and an aggregation of the weighted sum is taken, where the kernel specifies the weights. Figure 2.2 illustrates this. The convolution operation is employed as a spatial filter that extracts features in an image. Kernels could be defined manually to extract specific features, such as horizontal or vertical edges. In a convolutional neural network, however, the kernels contain weights that are learned during training. In this way, the neural network learns which features in the image are important.

Convolutional neural networks make use of convolution to impose certain properties on representations of data, such as equivariance to translation. These properties are especially useful for images, since images have a topological structure: they are represented by a grid of pixels. For each convolutional layer in a CNN, a neuron h is computed using only a locally contained subset of neurons in a previous layer, as illustrated in Figure 2.3. This subset of neurons is known as the local receptive field of h [26]. Local receptive fields are implemented by using a kernel for convolution that is smaller than the size of the input, and it leads to what is known as sparse connectivity in the layer.

Stellenbosch University https://scholar.sun.ac.za

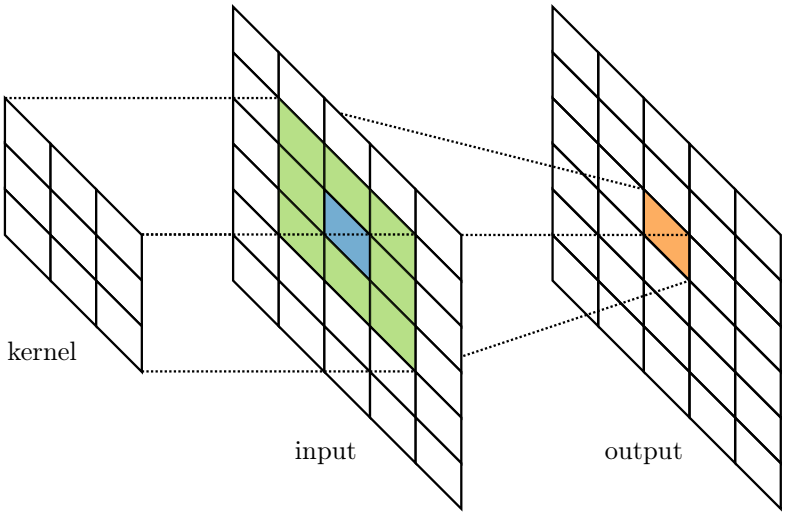


Figure 2.2: An illustration of two-dimensional convolution. Here the kernel is centred on the dark blue pixel. The kernel is multiplied elementwise with this region of the image and aggregated to the corresponding pixel of the output.

Sparse connectivity allows for a more efficient model as there are fewer computations and lower memory demands. Complex interactions between variables can then be efficiently described by using multiple layers of simple building blocks that are sparsely connected.

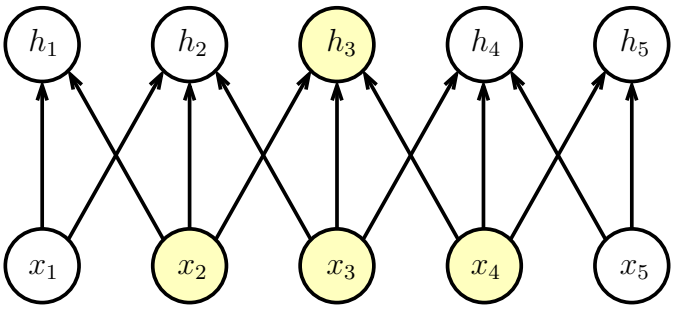


Figure 2.3: An example of a one-dimensional convolutional layer in a neural network. x_2, x_3, x_4 form the local receptive field of h_3. Redrawn from [23].

Another property of convolution as an operation in a neural network is parameter sharing. In a fully-connected neural network layer, there are distinct weights for each input of every neuron. However, in a convolutional layer, the set of weights in the kernel is applied to all receptive fields of the output.

It turns out that with these two properties alone, a CNN is many orders of magnitude more efficient than a fully-connected neural network for approximating a linear function to detect edges in an image [23]. Furthermore, we note that convolution is equivariant to translation. Convolution creates a map that shows where certain features appear in the input. If these features are translated in the input, the resultant feature map is

Stellenbosch University https://scholar.sun.ac.za

translated accordingly. This means that convolution can highlight (or detect) features regardless of their position in the input image.

For regularisation, that is to prevent the network from overfitting to the training data, we can use a pooling function that takes in the output of a convolutional stage. A pooling function is a summary statistic of a local neighbourhood in the input. A common type of pooling is max-pooling, where we take the maximum value of a region of prespecified size. If the pooling region is smaller than the input, then we achieve further efficiency through downsampling. Since a summary statistic is taken, the resolution is reduced. This means that the exact location of a feature becomes less important. As an additional regularisation method, we use dropout [27]. Dropout regularisation works by temporarily removing neurons and their connections from the network during training. Dropped neurons are usually chosen at random. Training a neural network with dropout results in sampling from an exponential number of different networks with reduced capacity. At test time, the predictions of these multiple networks are approximately averaged by a single network with dropout disabled and scaled weights [27].

Convolution introduces a prior on the weights of a neural network, dictating that the representation a layer learns should involve local interactions (local receptive fields) and should be equivariant to translation. Additionally, pooling introduces a prior that each unit in a layer should be invariant to small translations.

2.1.5 Pretrained neural networks

In this thesis we employ CNNs for visual relationship recognition. CNNs are comprised of convolutional and pooling layers, followed by at least one fully-connected layer. It is not always necessary to train such a network from scratch (i.e. with completely randomly initialised weights). When data is limited it is common practice to initialise a CNN with what is referred to as pretrained weights. These pretrained weights are typically trained on the ImageNet dataset [28], which contains over a million images from a thousand object classes. Then, randomly initialised fully-connected layers are appended to the pretrained network for a task in a target domain.

The process of using weights pretrained on data from a source domain and adapting them to a target domain is a form of transfer learning. There are two options to adapt the weights. The first is to freeze the weights of the pretrained section of the network and only train the newly appended layers. Earlier layers may extract more general features and so it might not be necessary to update their weights. In this way, the pretrained weights are being used as a fixed feature extractor and the appended layers may act as the classifier. The feature extractor thus produces a lower-dimensional representation of the input data, to be passed to the classifier. The second option is to also update (or finetune) the pretrained weights. Sometimes the second option can offer representations of the data that lead to improved performance in the target domain. For our visual relationship experiments, we make use of the first option.

Specifically, we take the convolutional base of the ResNet-18 model [29] as a pretrained feature extractor. ResNet-18 achieves a good balance between size (number of parameters) and performance. Its architecture was developed after observing that when a network is made deeper, accuracy saturates and then degrades due to numerical issues in

Stellenbosch University https://scholar.sun.ac.za

the computation of gradients during training. To overcome this, He et al. [29] use what is known as skip connections, as illustrated in Figure 2.4. These connections are identity mappings that get added to the output of deeper layers. In this way, stacked layers prior to the skip connection's point of connection learn a residual function. Skip connections do not add significant computational complexity nor additional trainable parameters to the network.

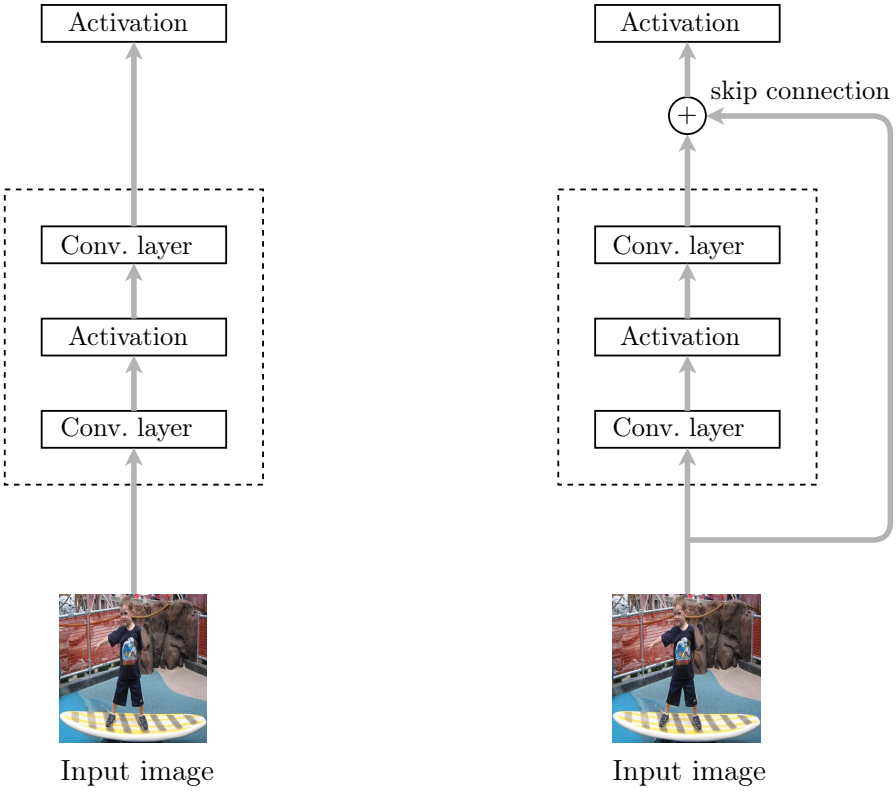


Figure 2.4: Left: a standard convolutional block. Right: a residual convolutional block with a skip connection. The convolutional block learns a residual mapping.

2.1.6 Cross-entropy loss

Another key component in representation learning is the objective, or loss function. A commonly used loss in classification is cross-entropy, since it is an objective that decreases when the model makes more correct than incorrect predictions. We can understand cross-entropy by taking an information theory perspective. For some probability distribution y, entropy is the expected amount of information in an event drawn from y [23]. Entropy also measures the expected number of bits needed to encode symbols drawn from y, under an optimal encoding. If we have access to y then an optimal encoding can be obtained by assigning $\log \frac{1}{y_i}$ bits to the i^{th} symbol. Taking the expectation of the number of bits

Stellenbosch University https://scholar.sun.ac.za

used yields what is called the entropy of y:

$$H(y) = \sum_i y_i \log \frac{1}{y_i} = -\sum_i y_i \log y_i. \tag{2.1}$$

So, if we have access to the distribution y, we can obtain an optimal encoding, and entropy is the lower bound on the expected number of bits used. Cross-entropy is the expected number of bits used for an encoding based on a distribution p, that is an approximation of the true distribution y:

$$H'(y) = -\sum_i y_i \log p_i, \tag{2.2}$$

where y_i is the true probability of the i^{th} symbol and p_i is the approximated probability. The goal then becomes to approximate the underlying distribution as closely as possible, while also minimising the expected number of bits in the encoding.

For multiclass classification, the distribution of labels for a given sample is of particular interest. In practice, y represents a ground truth label encoded as a one-hot vector. The softmax function is applied to the output of a neural network to obtain normalised class probabilities, p. The cross-entropy loss for a single sample is obtained by applying Equation 2.2. When employing mini-batch gradient descent to update the weights of the neural network, we take the mean loss over all samples within a mini-batch, and calculate gradients.

Cross-entropy is used to measure the difference between two distributions y and p, and minimising cross-entropy with respect to p is equivalent to minimising the Kullback-Leibler divergence between y and p [23]. Cross-entropy is thus minimised by making p closer to y, as can be seen in Figure 2.5. This seems like a reasonable objective for training and has proven its power in an abundance of scenarios [28; 30; 31; 29; 32]. The caveat is that when training a neural network, a large amount of balanced data is often required to obtain a reasonable estimate for y from any given input. The requirement for a large dataset is due to neural networks containing many parameters that must be learned. By balanced data, we mean that there are roughly the same number of instances of each class in the dataset. When this does not happen, as in a long-tailed class distribution, the contribution to the loss by rare classes is relatively low and estimating p in a manner that reliably models the rare classes becomes difficult.

2.1.7 Properties of representations

Neural networks create layered (or hierarchical) representations of data that can be used for tasks such as visual relationship recognition. A number of assumptions and properties are contained within these representations.

1. **Smoothness:** If two input samples x_1, x_2 are such that $x_1 \approx x_2$, then $f_w(x_1) \approx f_w(x_2)$. This assumption is helpful for generalisation, since we can say something about the representations of unseen points depending on their proximity to seen examples from the training set.

Stellenbosch University https://scholar.sun.ac.za

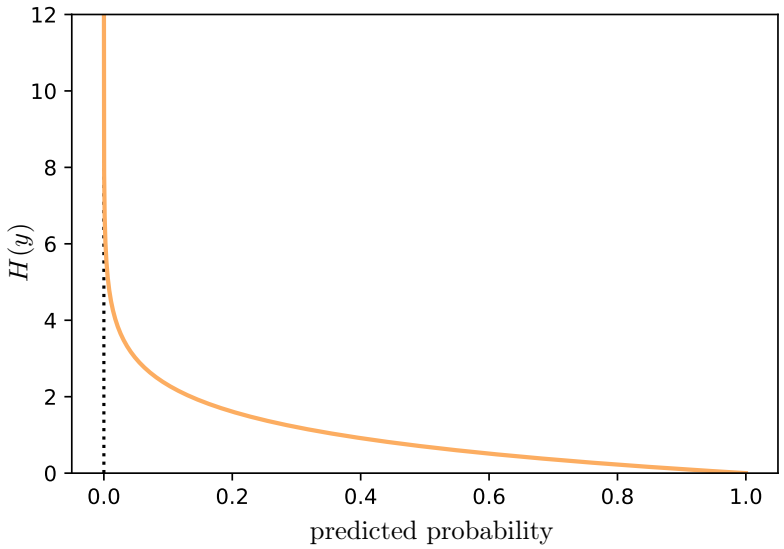


Figure 2.5: Plot of the cross-entropy when the true label is 1. It can be seen that the loss is minimised when predictions are closer to the true label.

2. **Multiple explanatory factors:** This is the assumption that the data is generated by multiple different underlying factors of variation. The objective then becomes to disentangle these factors.

3. **Depth:** The number of paths in a neural network grows exponentially with its depth. There are also some theoretical results that argue a deep representation is exponentially more efficient in the way that it re-uses features, than a shallow network [22].

4. **Hierarchical organisation of explanatory factors**: High-level concepts are built up from multiple low-level concepts. In the context of neural networks for image classification, the earlier layers might detect edges, while later layers might detect textures, patterns and objects [33]. It can also be said that these high-level concepts are abstractions of low-level concepts. Since abstract concepts can be more invariant to local changes of the input [22], ill-conditioned representations might be mitigated to some extent.

5. **Shared factors across tasks:** Sometimes, having a multitask training setup allows a model to leverage important information across tasks [6].

2.1.8 Learning a representation

In the case of neural networks, and for the task of image classification, representations are normally learned through the minimisation of the cross-entropy loss function. This often requires a training set of considerable size. A large training set is not always available, and there can be an imbalance in the number of samples per class which makes learning a representation even more challenging. Visual relationship recognition exhibits this problem: there is an extremely long tail in the distribution over possible classes,

Stellenbosch University https://scholar.sun.ac.za

resulting from a small number of classes that have many samples, while the vast majority has only a few or no samples. It may thus be desirable to learn representations that are robust to such a severe class imbalance. We can do this by injecting task specific priors into the architecture or into the loss function, but there is little evidence that this would generalise well. What might be more worthwhile is to consider architectures, learning objectives, and training procedures that more effectively learn representations of data to aid the task at hand.

In subsequent sections we formalise different training procedures, architectures and loss functions that we will investigate in the context of visual relationship recognition.

2.2 Class-selective batch construction

Neural networks trained with mini-batch gradient descent typically use a standard batching strategy. For each training iteration a mini-batch of some prespecified size is sampled without replacement, uniformly across all samples in the training set. For visual relationship recognition where the data is often heavily skewed, this standard approach to batch selection is likely to pick samples mostly from a small number of frequently occurring classes. The network may thus learn these dominant classes very well, but would be unable to recognise the vast majority of classes in the long tail of the data distribution.

In an effort to mitigate the potential problem with standard batching and expose the network to more classes in the tail of the training dataset, we implement the following batch construction strategy (as used by Schroff et. al [34]). For a particular visual relationship recognition task (which can be to predict the subject, or to predict the predicate, or to predict the object in a given image crop), we sample at every training iteration n classes from the vocabulary (of size N) of that task, uniformly at random. We then randomly select m samples from each of those n classes, for a mini-batch of size mn. Figure 2.6 illustrates this class-selective batch construction for $N = 6$, $n = 3$ and $m = 2$.

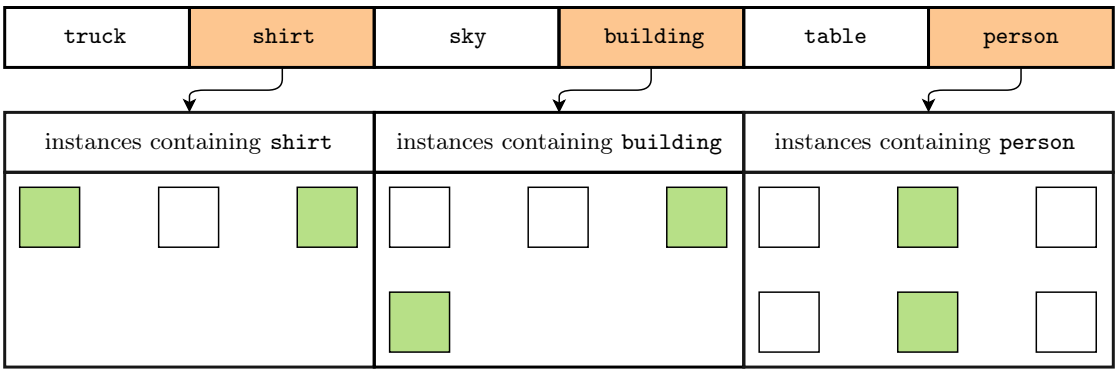


Figure 2.6: For a vocabulary of size $N = 6$, we randomly select $n = 3$ classes. Then, from each of the three classes, we randomly select $m = 2$ instances. Green boxes indicate all instances that are sampled for the construction of a single mini-batch.

Stellenbosch University https://scholar.sun.ac.za

Constructing batches in this manner would allow the network to learn from all the classes in a particular task, in equal measure. We hypothesise that it can lead to better performance on the many rare classes in the long tail of the data, potentially at the expense of reduced performance on the small number of dominant classes. Of course, there is now a risk of biasing the network against the true distribution of the data and impede its ability to generalise properly. We will investigate these issues experimentally.

A question arises: how many mini-batches should be constructed before the elements of all constructed mini-batches span the set of classes? We show the probability of each class appearing at least once after k mini-batches are constructed, for the worst case of $n = 1$. Consider the experiment of sampling (with replacement) a single class from the vocabulary, k times. For $i = 1, 2, \ldots, N$, let E_i be the event that class i appears at least once. The event that each class appears at least once is the intersection over the E_i's. By De Morgan's law and the inclusion-exclusion principle, we have

$$P\left(\bigcap_{i=1}^{N} E_i\right) = 1 - P\left(\bigcup_{i=1}^{N} E_i^c\right) = 1 + \sum_{i=1}^{N} (-1)^i \binom{N}{i} \left(\frac{N-i}{N}\right)^k. \tag{2.3}$$

Equation 2.3 provides a means of determining a confidence that all classes are being considered at least once during training. Figure 2.7 shows plots of this confidence for $N = 10$ and $N = 49$. It seems that one does not require an unfeasible number of mini-batches before being at least 99% confident in training on all classes. In the visual relationship recognition datasets considered later, however, there can be up to 100 classes for a single task. Equation 2.3 now presents some numerical challenges as the binomial coefficients become very large for $N > 50$ and small i. Consequently, we generate a plot by numerically determining the soughtafter probabilities. To do so, for varying values of k, we construct k mini-batches 500 times. The probability in Equation 2.3 can then be approximated by counting the fraction of times the elements of all k mini-batches span the set of classes. Figure 2.8 shows that one requires somewhere around 900 mini-batches before achieving a confidence close to 100%. Experiments in later sections create batches exceeding this mini-batch threshold. The analysis presented here is for the worst case where $n = 1$. We expect the mini-batch threshold to be much lower when $n > 1$ classes are sampled, as would typically be the case in practice.

2.3 Multitask learning

In addition to class-selective batch construction we also explore multitask learning, which can be thought of as an inductive form of transfer learning where knowledge is transferred across different tasks. More specifically, multitask learning makes the assumption that the predictive model should explain multiple tasks. This assumption can also be described as an inductive or learning bias [35]. The premise is that it might lead to a more robust model, capable of better generalisation [6].

Multitask learning, in the context of neural networks, changes the way data representations are learned by modifying the architecture. It works by using the domain information contained in the training signals of multiple related tasks as an inductive bias. Then, when using a shared representation to learn these tasks, information can be transferred

Stellenbosch University https://scholar.sun.ac.za

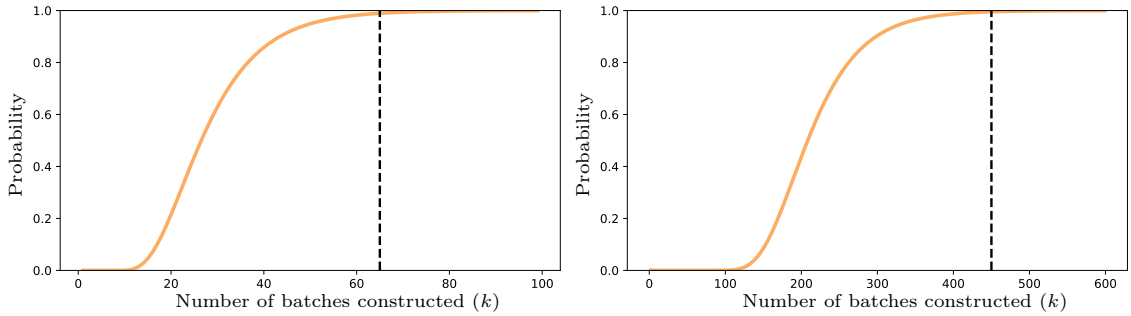


Figure 2.7: Left: probability of each of $N = 10$ classes appearing at least once after k mini-batches are constructed. At least 65 batches are required before being 99% confident that all classes appear at least once. Right: probability of each of $N = 49$ classes appearing at least once after k mini-batches are constructed. Here the 99% threshold for k is at 450. The black dashed line indicates this threshold.

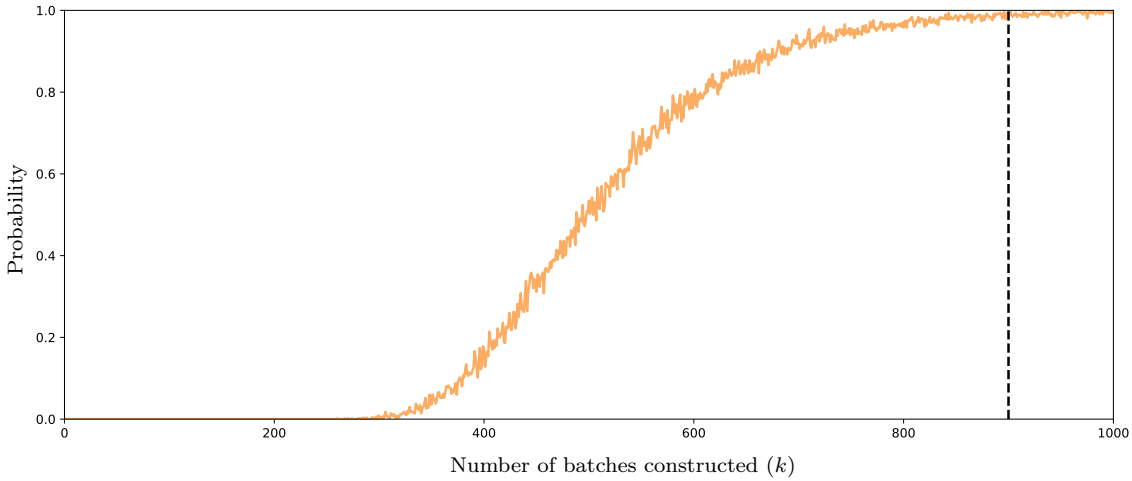


Figure 2.8: Numerically approximated plot of the probability that each of $N = 100$ classes appears at least once after k mini-batches are constructed. Here around 900 or more mini-batches are required before achieving a confidence close to 100%. The black dashed line indicates this threshold.

across tasks. Training signals refer to the errors that are accumulated when calculating gradients during a backward pass through a neural network.

Understanding task relatedness is important, since the efficacy of a multitask model might be predicted if task relatedness can be determined. Caruana [6] defines a few heuristics for relatedness, of which we discuss two related to our context.

1. Related tasks share input features. Since visual relationship recognition is treated as an image classification problem, the input features would be the union bounding box of the visual relationship.

2. Tasks can also be related when they share hidden representations. This can take the form of hard parameter sharing where hidden layers are shared among tasks. Or, it can take the form of soft parameter sharing where the distances between parameters of networks for each task are regularised to be similar. Hard parameter

Stellenbosch University https://scholar.sun.ac.za

sharing, however, is still a popular method of multitask learning [36] despite being introduced more than 20 years ago by Caruana.

Caruana [6] also suggests that even when tasks are related (as determined by the above definition), multitask learning may not offer improvements over single-task learning. Whether or not task relatedness can be taken advantage of depends on the learning algorithm [6].

There are a few reasons why multitask learning is often considered useful.

1. **Data amplification:** Even though the separate tasks share the same input features, there are more training signals when compared to the single-task setting. In our case, this occurs since we are minimising either three, four, or six loss functions with shared network layers or representations (as we explain in Section 2.5). Multiple training signals for the same input features act as a type of data amplification.
2. **Feature selection:** Since images are high-dimensional, it can be difficult for a model to distinguish between relevant and irrelevant features [36]. With multiple training signals, however, multitask learning contributes towards better feature selection due to data amplification.
3. **Representation bias:** Representation bias is perhaps the main driving factor behind multitask learning. Multitask learning introduces an inductive bias that favours a hypothesis (or model) explaining multiple tasks and has been shown to lead to better generalisation. In the context of visual relationship recognition, we define the prediction of each element of the (subject, predicate, object) triplet as a task. The hope is that using a shared representation, trained on all three tasks, may result in better recognition of the visual relationship triplet.
4. **Regularisation:** Training signals of different tasks have different noise patterns [36]. As a result, the learning procedure is likely to be regularised by the aggregation of multiple noise patterns.

Mini-batch gradient descent is a stochastic search procedure. Weights are randomly initialised, then after a forward pass of a batch of data, the gradient (of multiple loss functions) with respect to the weights are calculated. This allows for the weights to be updated with gradient descent so that the loss function moves towards some minimum. In multitask learning, the error gradients from multiple losses constructively and destructively interfere in the shared layers [6]. A shared representation that strongly favours a particular task at the expense of another will be influenced by mini-batch gradient descent to favour both tasks instead. The result is a shared data representation that favours multiple tasks.

2.4 Ranking loss

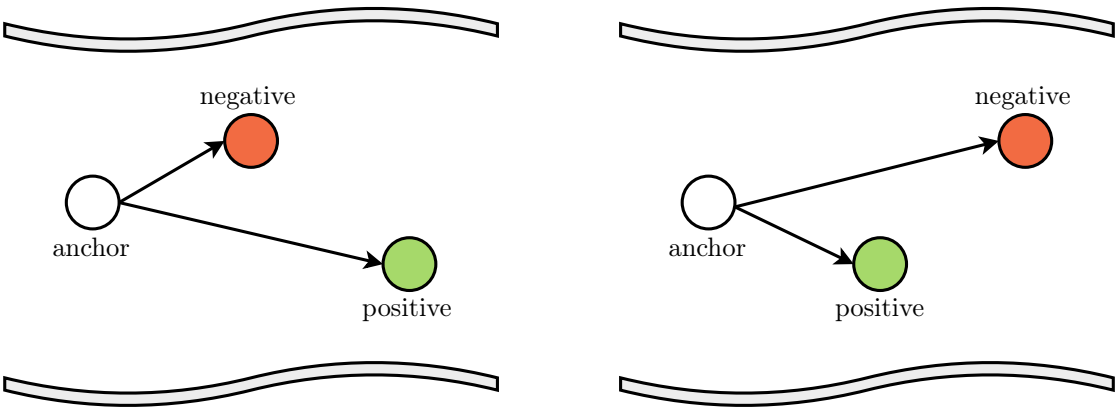
We also extend the multitask learning paradigm by including additional tasks with an objective other than cross-entropy. Specifically, an embedding space is learned by min-

Stellenbosch University https://scholar.sun.ac.za

imising a ranking[1] loss function [34]. Using a ranking loss imposes an additional property on the representations (or embeddings) to be learned. That is, in the embedding space, samples that are from the same class are closer together than samples from a different class. A classifier is trained jointly to output elements of the visual relationship triplet. In the extended multitask paradigm standard batching or class-selective batch construction strategies can be employed.

The idea behind the ranking loss function is to map a pair of similarly labelled samples to points on the output manifold that are relatively closer than a pair of dissimilar samples as illustrated in Figure 2.9. In our case similar samples are those that share the same class and dissimilar samples are from different classes, but the idea could be extended to include other notions of similarity. We use the Euclidean distance as a measure of similarity on the output manifold. For input samples $x_i, x_j \in \mathbb{R}^q$ and a neural network f_w we have

$$D_{ij} = ||f_w(x_i) - f_w(x_j)||_2^2. \tag{2.4}$$



a) Embedding before minimisation of the ranking loss function.

b) Embedding obtained after the minimisation of the ranking loss function.

Figure 2.9: Minimisation of the ranking loss moves the positive sample closer to the anchor and the negative sample further away.

The ranking loss as we implement it is a function that accepts three inputs. For an arbitrary sample x_a (called the anchor), a positive sample x_p and a negative sample x_n are selected as inputs to the ranking loss. A positive sample is one that shares its class label with the anchor while a negative sample is of a different class label. A collection of three such samples will be referred to as a trio[2]. Figure 2.10 shows two example trios. To obtain an embedding where similar samples are closer to one another than dissimilar samples, we must have that

$$D_{ap} + m < D_{an}, \tag{2.5}$$

[1] Schroff et al. [34] use the term triplet loss, which is a specific type of ranking loss. We use the more general term to avoid confusion with the word "triplet" in visual relationship triplet.

[2] It is more standard to refer to such a collection of three samples as a "triplet", but in this thesis we use the word "trio" again to avoid confusion with the notion of a visual relationship triplet.

Stellenbosch University https://scholar.sun.ac.za

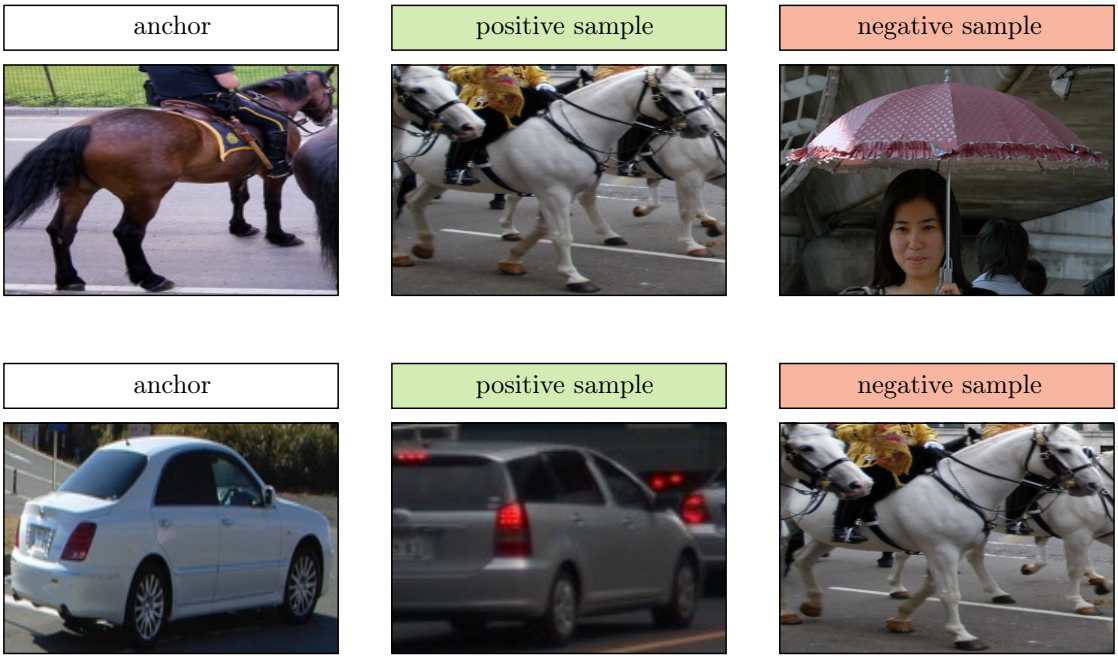


Figure 2.10: Top: for an anchor with the object label `horse`, a positive sample with the same label, and a negative sample with a different label (`umbrella`) are selected to form a trio for the ranking loss. Bottom: for an anchor with the label `car`, a positive sample with the same label, and a negative sample with a different label (`horse`) are selected.

where m is a specified margin, typically tweaked as a hyperparameter. This requirement states that the distance between the anchor and a negative sample must be greater than the distance between the anchor and a positive sample, by some margin. Since relative distances are of interest, sample trios which already meet the margin requirement need not be considered. It is more important to minimise the loss over sample trios that violate the constraint in Equation 2.5. With that in mind, the ranking loss over the three samples is defined as:

$$\mathcal{L}_{trio}(x_a, x_p, x_n) = \max(0, D_{ap} - D_{an} + m). \tag{2.6}$$

The complete loss function is the aggregation of the one in Equation 2.6 over all sample trios in the dataset:

$$\mathcal{L} = \sum_{\forall(a,p,n)} \mathcal{L}_{trio}(x_a, x_p, x_n). \tag{2.7}$$

Note that the ranking loss does not explicitly encourage the distances between positive samples to approach zero. It instead attempts to keep all positives closer than any negatives for each example. This means that there is no constant margin for all negative samples. A constant margin is less desirable because it might embed visually diverse classes in a similarly small space as visually coherent ones.

Stellenbosch University https://scholar.sun.ac.za

2.4.1 Training data distribution search

There are important computational considerations with the ranking loss. The number of sample trios of the form (anchor, positive, negative) grows cubically with the size of the training set and quickly becomes computationally unfeasible. However, it might not be necessary to consider all trios, which prompts the use of sampling techniques. For each positive sample an appropriate negative sample needs to be chosen, but this can also be difficult. Here is the crux: the ranking loss incurs a value of zero very quickly after a few training iterations as most negative samples do not contribute to the loss. In fact, for randomly sampled embeddings on the unit hypersphere, it is likely to obtain embeddings that are a distance of $\sqrt{2}$ apart [37]. This means that if our margin m is less than $\sqrt{2}$ and our embeddings are constrained to the unit hypersphere, then we would have many samples that do not contribute to the loss.

We may distinguish between the following three types of negative samples.

1. **Easy negatives:** sample trios where $D_{ap} + m < D_{an}$. The ranking loss thus incurs a value of zero and weights are not updated.
2. **Hard negatives:** sample trios where $D_{an} < D_{ap}$. Here the negative sample is closer to the anchor than the positive sample. This typically leads to collapsed models (the neural network learns a constant function) where training converges to a bad local minimum [34] due to a gradient with high variance and low signal-to-noise ratio [37].
3. **Semi-hard negatives:** sample trios where $D_{ap} < D_{an} < D_{ap} + m$. Here the negative is not closer to the anchor than the positive but still gives a positive loss and thus will lead to an update in the weights of the network.

Figure 2.11 illustrates the three kinds of negative samples. Sampling semi-hard negatives is often also called semi-hard negative mining. We may view semi-hard negative mining as a type of training data distribution search.

2.4.2 Offline vs online mining of samples

There are, broadly speaking, two ways of sampling negatives. The first is offline mining, where negative samples are generated at regular time intervals (the start of each epoch, say). At such a time interval, embeddings for the entire training set are computed. Then, for each positive pair a semi-hard negative sample is randomly selected to form a trio. This can be inefficient and if mini-batch gradient descent is used, then the weights of a neural network are updated more frequently than the training set embeddings. Such a strategy leads to outdated embeddings.

Alternatively, we can make use of online mining. Instead of computing embeddings over the entire training set, embeddings are computed per batch. Then, for each positive pair within a batch, a semi-hard negative sample is randomly selected to form a sample trio.

Stellenbosch University https://scholar.sun.ac.za

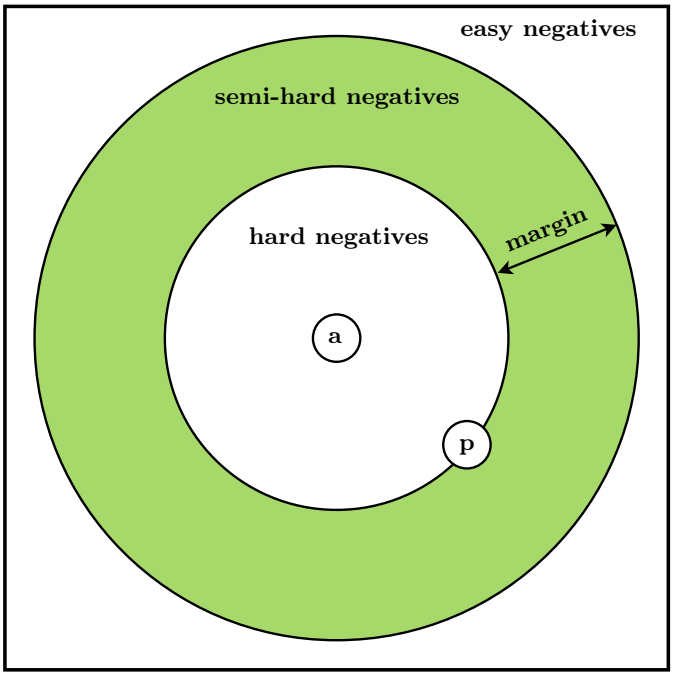


Figure 2.11: A representation of where negative samples are located relative to the anchor a and the distance to the positive sample. The green band is the sweet spot defined as semi-hard negatives, and negatives in this region should be sampled during training.

2.4.3 Siamese network architecture

The ranking loss works on a trio of samples. Consequently, the neural network needs to accept more than one input. Siamese networks [38] (as we use them) are comprised of multiple copies of a neural network f_w. The copies share the same set of weights and receive three images x_a, x_p and x_n as input. The goal is then to produce representations $f_w(x_a)$, $f_w(x_p)$ and $f_w(x_n)$ such that $f_w(x_a)$ and $f_w(x_p)$ are close in proximity (since x_a and x_p belong to the same class), while $f_w(x_a)$ and $f_w(x_n)$ are more distant (since x_a and x_n are from different classes). Figure 2.12 illustrates such a network. The choice of neural network architecture depends on the problem definition and form of data. We are working with image data, so a natural choice is a convolutional neural network.

As with any neural network, Siamese networks are functions that compute representations of the input data. There is a notion that Siamese networks learn semantic similarity between objects, since we define a distance metric. In that sense, training a Siamese network can be classified as metric learning. Applications of metric learning are visual tracking [39; 40], person re-identification [41], facial recognition [34], and signature verification [38], where there is typically little data to learn from. Since metric learning is used in domains where there are often few examples per class (like one-shot learning), we hypothesise that Siamese networks may perform well in the visual relationship recognition task, with its long-tailed distribution over class labels.

Stellenbosch University https://scholar.sun.ac.za

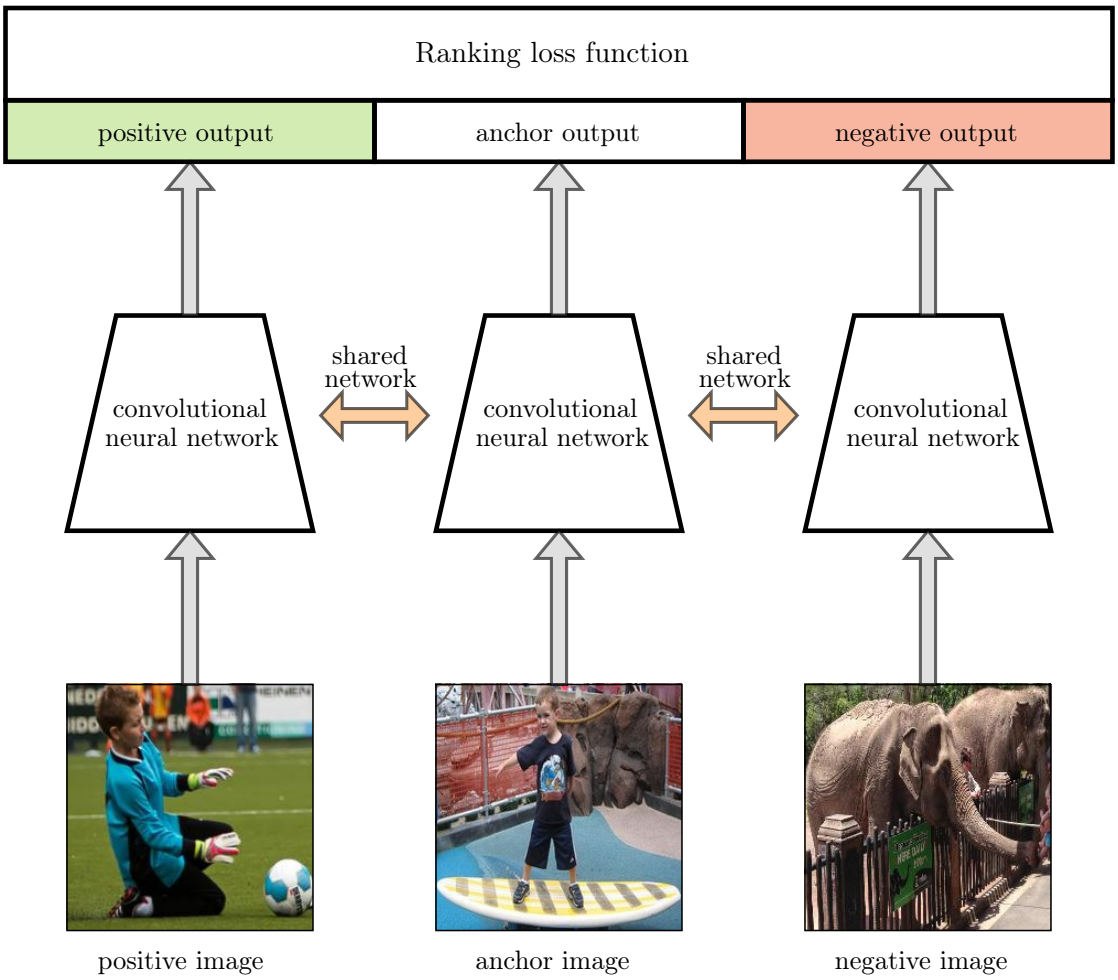


Figure 2.12: Siamese network for the task of subject prediction. Here the anchor image contains `boy`, the positive image contains `boy` and the negative image contains `elephant`. For each input image, an embedding is calculated using a shared network and passed to the ranking loss.

2.5 Models for visual relationship recognition

This section discusses how we apply the aforementioned strategies in the context of visual relationship recognition. The goal is to train a network that takes an image as input, cropped around a pair of objects, and outputs a (subject, predicate, object) triplet. Training labels are used to define fixed vocabularies for each element of the visual relationship triplet. We may therefore treat visual relationship recognition as classification, and networks are set up to output normalised class scores over triplets. We note that subjects and objects usually share a vocabulary, but it is not a strict requirement.

As mentioned, instead of attempting to train a convolutional neural network to output one massive vector of scores over all possible triplets, we consider three separate tasks: predicting the subject label, predicting the predicate label, and predicting the object label. This simple strategy already deals with the combinatorial challenge (the high

Stellenbosch University https://scholar.sun.ac.za

number of possible triplets). Each of the three separate tasks has far fewer possible classes, and by making the simplifying assumption that the tasks are conditionally independent given an input image, we may combine their normalised output scores through simple multiplication. The top scoring triplet can then be obtained by combining the top scoring elements from each of the three tasks.

2.5.1 Single-task learning

Three separate neural network models are created to predict the subject, the predicate and the object from the same image crop. Each network consists of the convolutional block of a pretrained ResNet-18 network, followed by three trainable, 2,048-dimensional fully-connected layers and a softmax output layer. Refer to Figure 2.13.

Single-task learning architectures will be trained with both standard batching and class-selective batch construction strategies.

2.5.2 Basic multitask learning

We use the convolutional base of ResNet-18, add two trainable, 2,048-dimensional fully-connected layers, and then split the network to three parts. Each part has its own 2,048-dimensional layer and a softmax output over the subjects, predicates and objects, respectively. Figure 2.14 clarifies. The first two fully-connected layers are thus shared and might learn effectively from the three different tasks. The network is trained to minimise the sum of cross-entropy losses over the three output vectors, using mini-batch gradient descent.

In multitask learning it is common to define a main task together with auxiliary tasks which could be less important. For our visual relationship recognition model we may want to regard each of the three tasks as equally important. However, when coupled with class-selective batch construction (as described in Section 2.2), we have to sample the m classes from a single task's vocabulary, at every training iteration, and then use the triplets from the complete labels of the training samples in the batch. We will experiment with how performance of the multitask model changes depending on which task is used for batch construction.

2.5.3 Hierarchical multitask learning

The implementation of our hierarchical multitask model is similar to the one described in Section 2.5.2. The difference is that an embedding layer is added to every branch of the multitask network, trained with the ranking loss function. Refer to Figure 2.15. Labels from each visual relationship task, in each batch, are used to construct sample trios in an online manner for each task. Gradients are then accumulated at every branch and errors are backpropagated during mini-batch gradient descent. Each task-specific branch of the network is shared between two tasks with different objectives (cross-entropy on the output scores and a ranking loss on the embeddings), while the middle layers are trained over all six objectives. In this way, there are hierarchies over multiple tasks. Note that

Stellenbosch University https://scholar.sun.ac.za

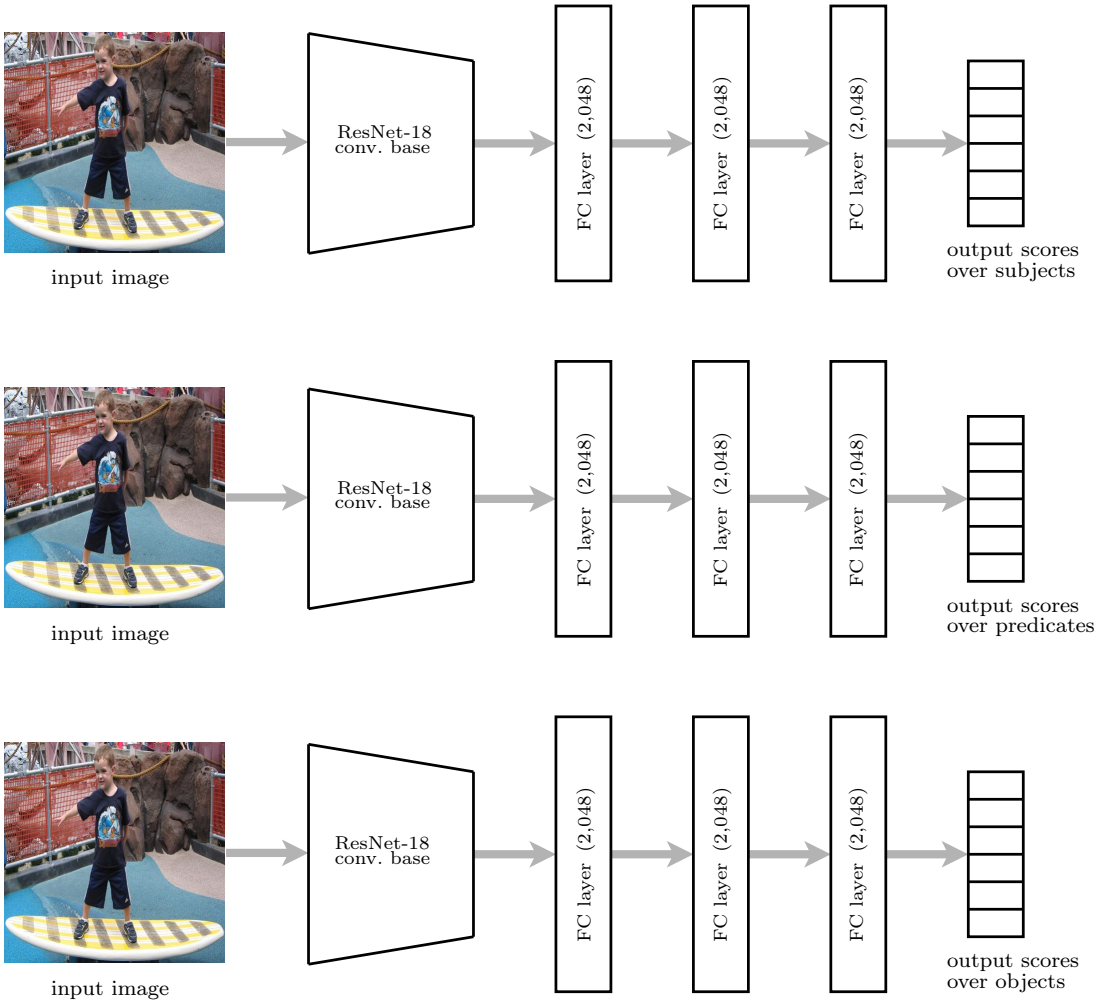


Figure 2.13: For single-task learning we construct three separate models to predict respectively the subject, predicate and object from a given image crop. The trainable fully-connected layers all have 2,048 neurons and the output is a softmax over the classes of each task. Dropout regularisation is used in the fully-connected layers.

the additional tasks act as a regulariser, and dropout is omitted in the trainable layers of this network.

2.5.4 Split-multitask learning

Instead of appending the embedding layers to the end of the neural network, they can be appended at the end of the shared representation (middle layers), as shown in Figure 2.16. Contrary to hierarchical multitask learning, split-multitask learning splits the objectives at the shared representation earlier in the network. This setup trains only the shared representations using six objectives. The task-specific branches in later layers of the networks then act as a classifier for each task. All layers are trained jointly, in an end-to-end manner. This network can also be trained in a two-stage approach, where the shared representation is first trained with the ranking loss and then frozen. Thereafter, a

Stellenbosch University https://scholar.sun.ac.za

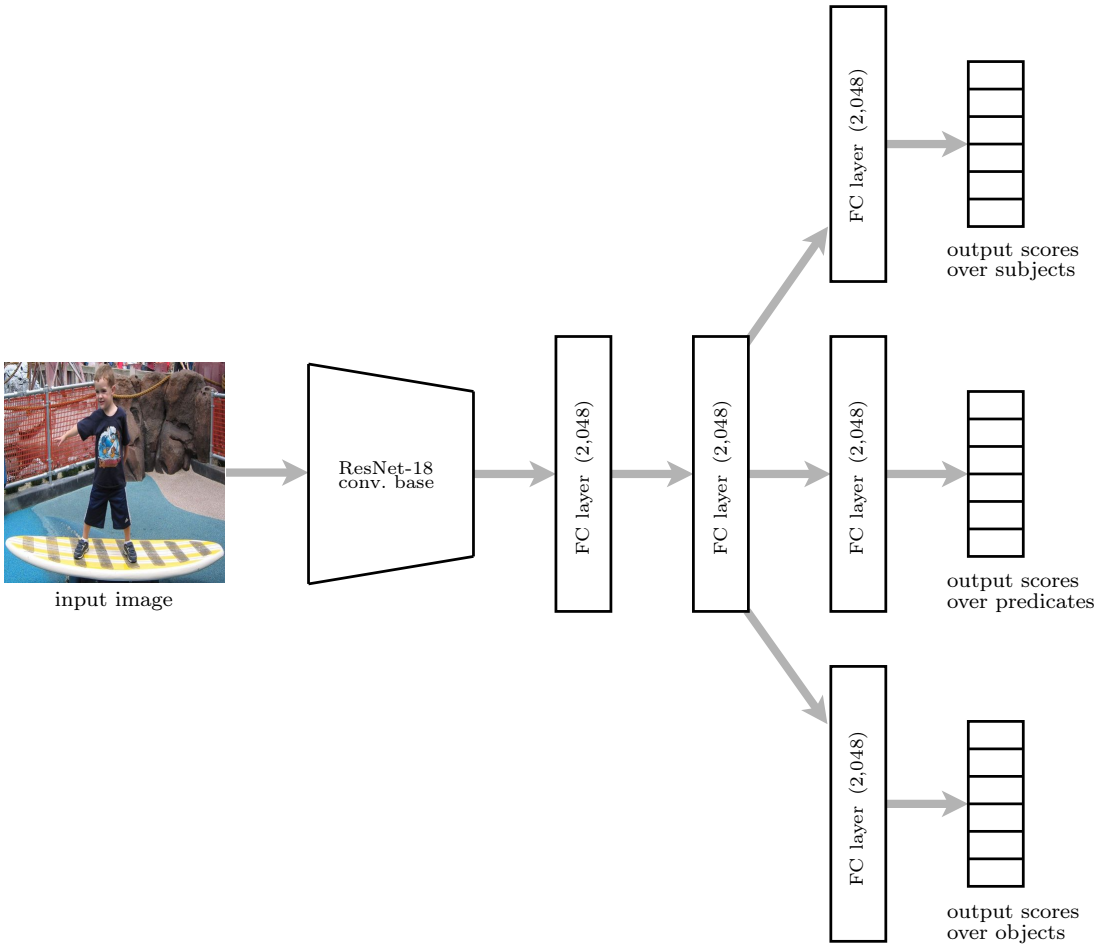


Figure 2.14: For the basic multitask learning setting, we construct a single model to output three score vectors over the subject labels, predicate labels and object labels. The shared and separate fully-connected layers all have 2,048 neurons and the outputs all use softmax. Dropout regularisation is used in the fully connected layers.

multitask classifier can be learned using the shared representation. End-to-end training tends to be preferred in literature, and our own informal experimentation showed that the two-stage approach yields inferior performance on the rare classes in the long-tailed distribution of visual relationships.

Stellenbosch University https://scholar.sun.ac.za

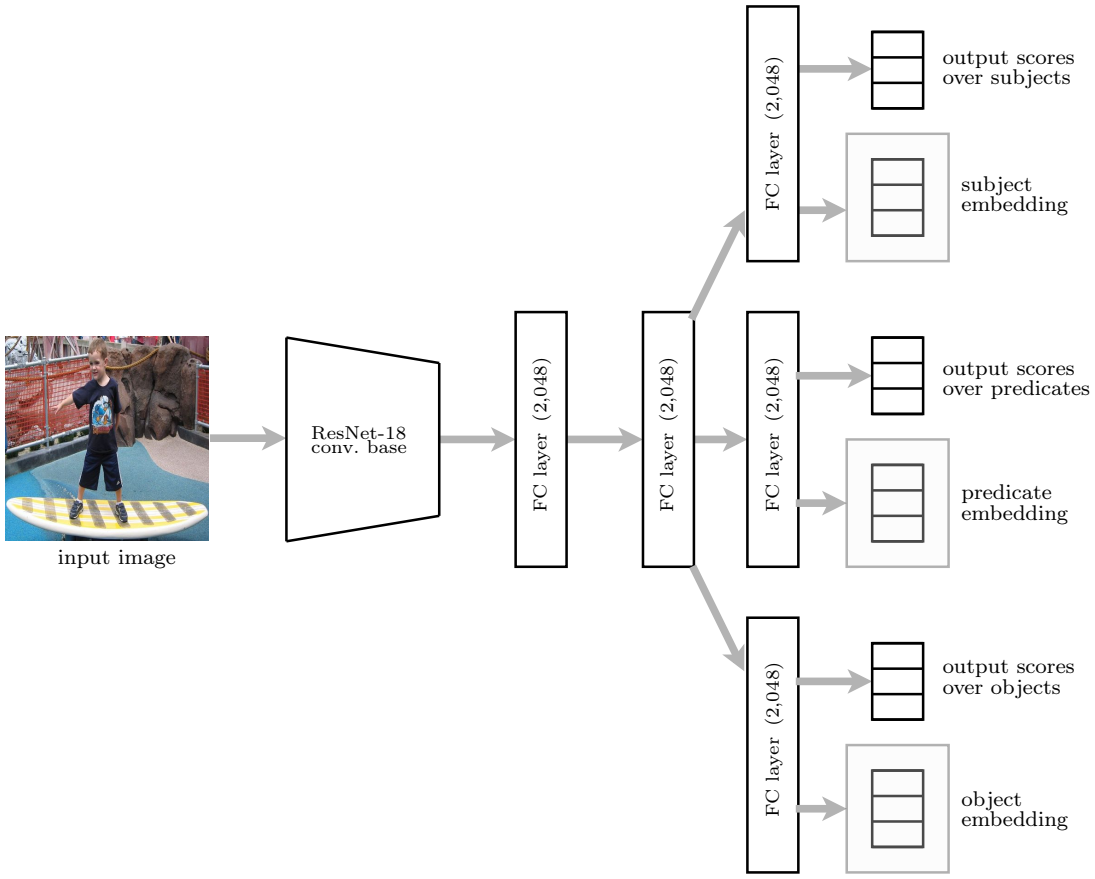


Figure 2.15: For the hierarchical multitask learning setting, we construct a single model to output three score vectors over the subject labels, predicate labels and object labels as well as three embedding vectors for each of the three tasks. That is, three ranking loss functions (one for each visual relationship task) are employed for training each embedding. The shared, separate, and embedding layers are all fully-connected and have 2,048 neurons, and the classification outputs all use softmax. The grey boxes contribute to the ranking loss function.

Stellenbosch University https://scholar.sun.ac.za

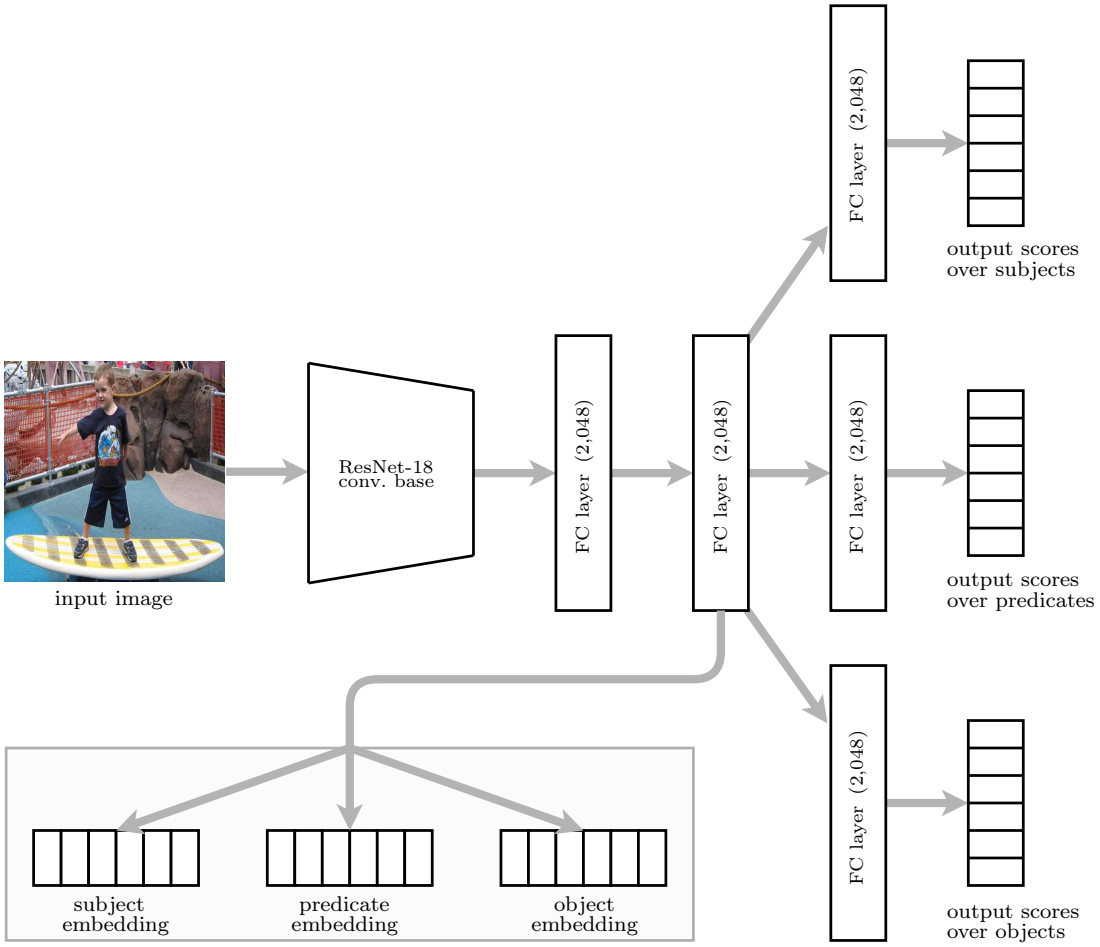


Figure 2.16: For the split-multitask learning setting, we construct a single model to output three score vectors over the subject labels, predicate labels and object labels as well as three embedding vectors for each of the three tasks. Now, instead of adding the embedding layers at the end of the network, we add them at the end of the shared layers. This is arguably closer to the standard definition of multitask learning since a single shared representation is being used. The shared, separate, and embedding layers are all fully-connected and have 2,048 neurons, and the classification outputs all use softmax. The grey box contributes to the ranking loss function.

Stellenbosch University https://scholar.sun.ac.za

Chapter 3

Datasets, metrics and preliminary experiments

This chapter introduces the dataset and performance evaluation metrics used for our experiments with visual relationship recognition. The challenges introduced in Chapter 1 will now become concrete through the nature of the dataset. Commonly used metrics are considered as well as metrics that are more informative with respect to performance on the long tail. Finally, and before training our models for the problem of visual relationship recognition, we test the ideas introduced in the previous chapter on the MNIST and CIFAR10 datasets.

3.1 Data

We make use of the VRD dataset of Lu et al. [4]. It contains 5,000 images, and a total of 37,987 relationship instances (triplets) that we split into a training set and a test set. Figure 3.1 illustrates a few visual relationships and Figure 3.2 represents an image from VRD, that we adapt for our visual relationship recognition models. In an effort to ensure that all classes are represented in both the training and test set, the visual relationships are split with respect to the predicate. For each predicate class, 80% of relationships containing that predicate are selected randomly and put in the training set. The test set is composed of the remaining 20%. The predicate is chosen since it has fewer samples per class in the tail and as such, any other choice is likely to result in the predicate's classes not being represented in either the training or the test set.

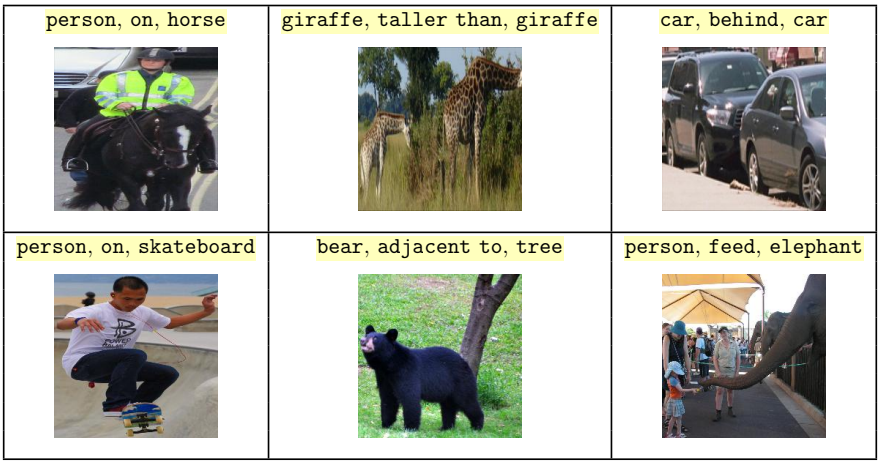


Figure 3.1: Some examples and their ground truth relationship from the VRD dataset.

Stellenbosch University https://scholar.sun.ac.za

a) `horse`, `next to`, `horse` b) `person`, `on`, `horse` c) `person`, `ride`, `bicycle`

Figure 3.2: Top: example image and visual relationship bounding boxes from the VRD dataset. Bottom: cropped images that our models accept as input, with ground truth labels.

Figure 3.3 illustrates the categories of predicates that exist in this dataset. Each predicate is an action verb (e.g. `kick`), a non-action verb (e.g. `wear`), a spatial relationship (e.g. `on top of`), a preposition (e.g. `with`), or comparative (e.g. `taller than`). The examples in this section also demonstrate the ambiguity that may exist in visual relationship labels. A labelled relationship can easily be replaced by a different label and still be semantically correct. For example, (`elephant`, `taller than`, `person`), in Figure 3.3 could have the predicate replaced by `next to`. In Chapter 4 we will qualitatively investigate how the models respond to this type of ambiguity in the dataset.

There are 100 labels shared between subjects and objects, and 70 labels for predicates, for a total of 700,000 possible (subject, predicate, object) triplet labels. All the labels of the individual elements are listed in Figure 3.4. We note that our training set contains only 15,448 of the 700,000 unique triplets. However, the manner in which the models are set up to output subject, predicate and object separately, potentially enables the recognition of triplets never seen during training.

The long-tailed nature alluded to throughout this thesis exists in this dataset not only at the relationship triplet level, but also at the level of subjects, predicates and objects, as shown in Figure 3.5. The figure also demonstrates the fact that the predicates have fewer samples per class in the tail, compared to the subject and object. We note that the VRD dataset is heavily biased towards the `person` class in the subject label. This is represented by the exceptionally large number of instances of class 0 in the subject

Stellenbosch University https://scholar.sun.ac.za

graph.

action	spatial	preposition	comparative	verb
person kick ball	person on top of ramp	motorcycle with wheel	elephant taller than person	person wear shirt

Figure 3.3: Five categories of predicates: action, spatial, preposition, comparative and verb. Green and red bounding boxes are around subjects and objects respectively. Note that predicates like `taller than` can be replaced by `next to` and still be semantically correct, demonstrating the ambiguity in visual relationships.

Subject and object labels					
person	sky	building	truck	table	shirt
chair	car	train	glasses	tree	boat
hat	trees	grass	pants	road	motorcycle
jacket	monitor	wheel	umbrella	plate	bike
clock	bag	shoe	laptop	desk	cabinet
counter	bench	shoes	tower	bottle	helmet
stove	lamp	coat	bed	dog	mountain
horse	plane	roof	skateboard	traffic light	bush
phone	airplane	sofa	cup	sink	shelf
box	van	hand	shorts	post	jeans
cap	sunglasses	bowl	computer	pillow	pizza
basket	elephant	kite	sand	keyboard	plant
can	vase	refrigerator	cart	skis	pot
surfboard	paper	mouse	trash can	cone	camera
ball	bear	giraffe	tie	luggage	faucet
hydrant	snowboard	oven	engine	watch	face
street	ramp	suitcase			

Predicate labels					
on	wear	has	next to	sleep next to	sit next to
stand next to	park next	walk next to	above	behind	stand behind
sit behind	park behind	in the front of	under	stand under	sit under
near	walk to	walk	walk past	in	below
beside	walk beside	over	hold	by	beneath
with	on the top of	on the left of	on the right of	sit on	ride
carry	look	stand on	use	at	attach to
cover	touch	watch	against	inside	adjacent to
across	contain	drive	drive on	taller than	eat
park on	lying on	pool	talk	lean on	fly
face	play with	sleep on	outside of	rest on	follow
hit	feed	kick	skate on		

Figure 3.4: List of subject, predicate and object labels, in no particular order. For brevity, labels like "on the top of" are shortened to "on top of" in this thesis.

Stellenbosch University https://scholar.sun.ac.za

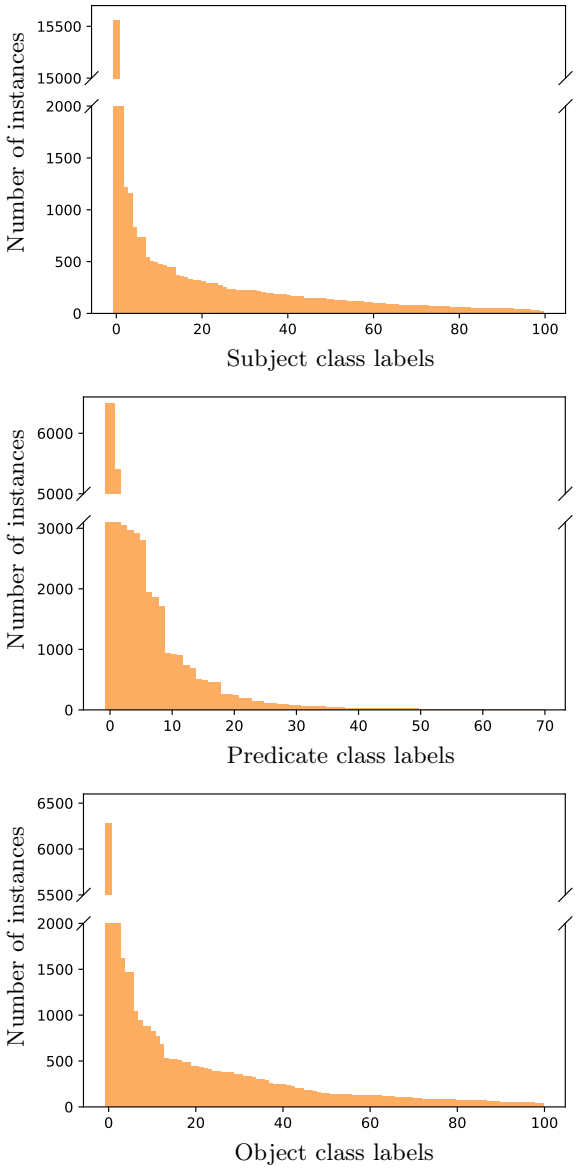


Figure 3.5: The distributions of subject, predicate and object class labels across the entire VRD dataset.

3.2 Evaluation metrics

The performance of the various models will first be evaluated in terms of predicting each of the three elements of a visual relationship, then in terms of predicting the triplet as a whole.

A standard metric for visual relationship recognition is recall-at-k, abbreviated as R@k and sometimes called the top-k accuracy, which measures the fraction of times the correct label occurs in the top k predictions (if ordered by output scores). For the tasks of

Stellenbosch University https://scholar.sun.ac.za

predicting individual elements, i.e. looking only at the output over subjects, or over predicates, or over objects, we will measure R@1 and R@3 on the test set. For the task of predicting the full (subject, predicate, object) triplet, we will measure R@50 and R@100 which seem to be standard practice for a label set of this size [4; 7; 9; 11]. Keep in mind that there are 700,000 possible triplets that can be predicted. We found that a random classifier yields an R@100 score of approximately 0.026% on the skewed test set.

In order to evaluate how effectively each model deals with the many rare classes in the tail of the data distribution, the mean per-class accuracy, abbreviated as MPCA, will also be measured over the test set. It effectively ignores class imbalance. Note that this metric is only used to evaluate the prediction of single elements (subjects, predicates or objects), and not the prediction of full triplets. The large number of possible triplets, and the fact that relatively few of them appear in the test set, make MPCA less informative in the case of full triplet prediction.

For an indication on how the models fare on only the rare visual relationship classes, a subset of the test set is constructed by keeping only those triplets for which the subject, predicate and object each have fewer than 1,000 instances across the full dataset (refer to Figure 3.5). Recall-at-50 and -100 over this subset will be referred to as tail R@k. We use counts over the full dataset as a proxy for rarity in the test set, and remind the reader that elements in the test set are distributed similarly to those in the full set.

3.3 Preliminary experiments

The models introduced in Chapter 2 are first tested on simpler datasets to get an indication of their behaviour on image classification. These experiments provide some evidence whether the techniques may deal with the challenges in the more complex task of visual relationship recognition.

3.3.1 Experimental design

The following set of steps describe the experiments that are discussed in this section.

1. We consider the MNIST [42] and CIFAR10 [43] datasets, which are both 10-class image classification problems.

2. We create subsets of these datasets, for various levels of non-uniformity in the distribution of classes (as illustrated Figure 3.6). The idea is to start with a uniform distribution (a balanced dataset) and gradually progress towards a long-tailed distribution. In each of the subsets, the total number of samples is kept constant with the minimum number of samples per class set to 20. The distribution of a subset is determined by $n_c \propto e^{-ac}$, where a controls the level of non-uniformity, n_c is the number of samples in class c (scaled to keep the total number of samples constant for different values of a), and $c = 0, 1, \ldots, 9$.

3. We train models using a pretrained network called AlexNet [28] as initialisation and then finetune. For each dataset we consider the following settings:

Stellenbosch University https://scholar.sun.ac.za

a) single-task standard batching (ST-SB) vs class-selective batch construction (ST-BC), with the cross-entropy loss function;

b) online ranking loss with standard batching (Ranking-SB) vs class-selective batch construction (Ranking-BC);

c) standard batching with the cross-entropy loss (ST-SB) vs the online ranking loss (Ranking-SB);

d) class-selective batch construction with the cross-entropy loss (ST-BC) vs the online ranking loss (Ranking-BC).

4. For each setting we evaluate performance using the mean per-class accuracy on a hold-out test set whose level of non-uniformity in its distribution over classes is similar to the training set used.

Recall that the ranking loss is not a classification loss, and as such we train an SVM classifier on the embeddings learned from the ranking loss. We make use of an SVM only for this simplified setting, and for visual relationship recognition we optimise the ranking loss in a joint, end-to-end training setting.

The experiment described above is repeated 42 times for each dataset and level of non-uniformity in Figure 3.6, to obtain an average MPCA as well as standard deviation in the MPCA across the runs. The standard deviation can give an indication as to how stably learning converges. For each run, the classes are shuffled so that a random class represents the majority class.

There are two questions of interest. The first is whether our strategy of class-selective batch construction performs better on the tail of the distribution when compared with standard batching. The second is whether the ranking loss creates representations that improve classification when paired with a classifier. Due to the nature of the experiment, the total number of samples are quite low (6,000) and will result in lower accuracy scores compared to using the full dataset.

3.3.2 Results and discussion

The discussion starts with the relatively simple MNIST dataset. MNIST is a database of handwritten digits, each in the form of a 28×28 greyscale image. There are only 10 classes (the digits from 0 to 9) and the intra-class variance is not that high. It is commonly used as a low benchmark for image classification, with a training and test set containing 60,000 and 10,000 samples respectively. CIFAR10 is a more complex dataset, also with 10 classes. It contains colour images of dimension 32×32, each belonging to either a type of vehicle or a type of animal. Since CIFAR10 is more complex, and the experiments contain a relatively small number of samples in the training set, we expect lower performance than on MNIST. Figure 3.7 shows a few examples from the MNIST and CIFAR10 datasets.

Stellenbosch University https://scholar.sun.ac.za

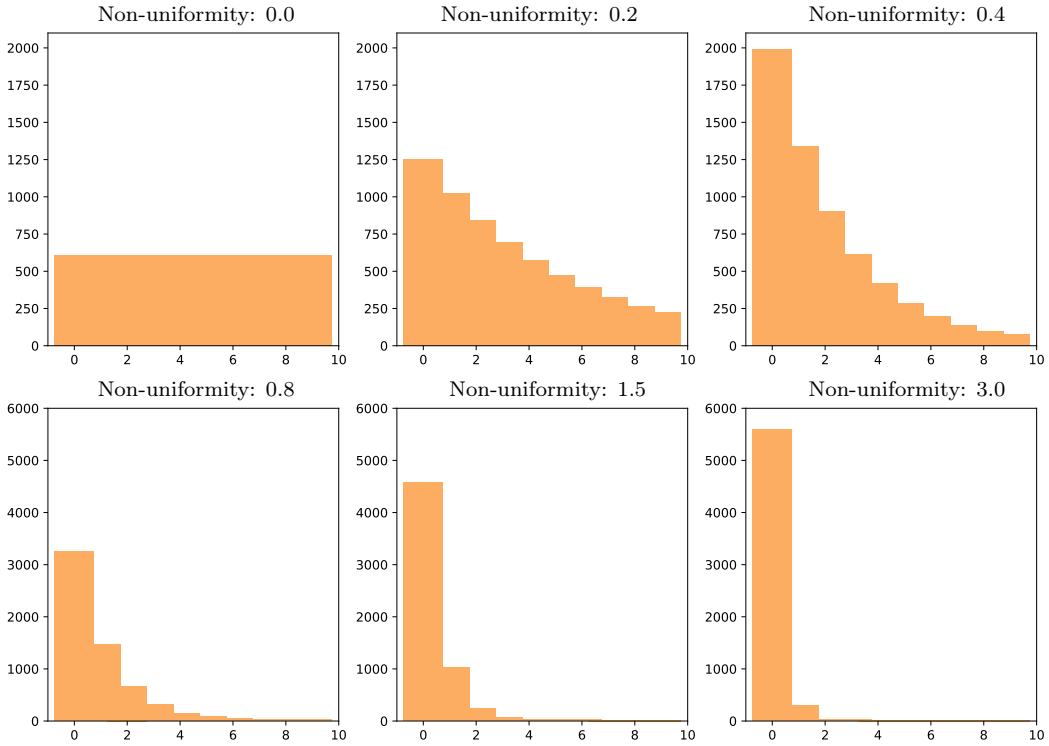


Figure 3.6: Class label distributions used for the preliminary experiments in this section, for various levels of non-uniformity. Note that the uniformity decreases as the level increases. For a level of 0, we have a uniform distribution with around 600 samples per class and a total of 6,000 samples. This total is kept constant across the different levels.

Figure 3.7: Top: five random examples from the MNIST dataset. Bottom: five random examples from the CIFAR10 dataset.

MNIST

Results for settings 3a and 3b (as listed in Section 3.3.1) are presented in Figure 3.8. For both loss functions, class-selective batch construction provides better performance

Stellenbosch University https://scholar.sun.ac.za

on the tail of the distribution, even for the most extreme case of non-uniformity. This is evident from the slower decrease in the curves with batch construction. It may be interesting that in the case of the cross-entropy loss function, batch construction has a low, fairly constant variance across runs. Batch construction might result in more stable training when dealing with a long tail setting in relatively simple classification problems. In contrast, the variance increases as the level of non-uniformity increases in the standard batching case, with cross-entropy as well as the ranking loss.

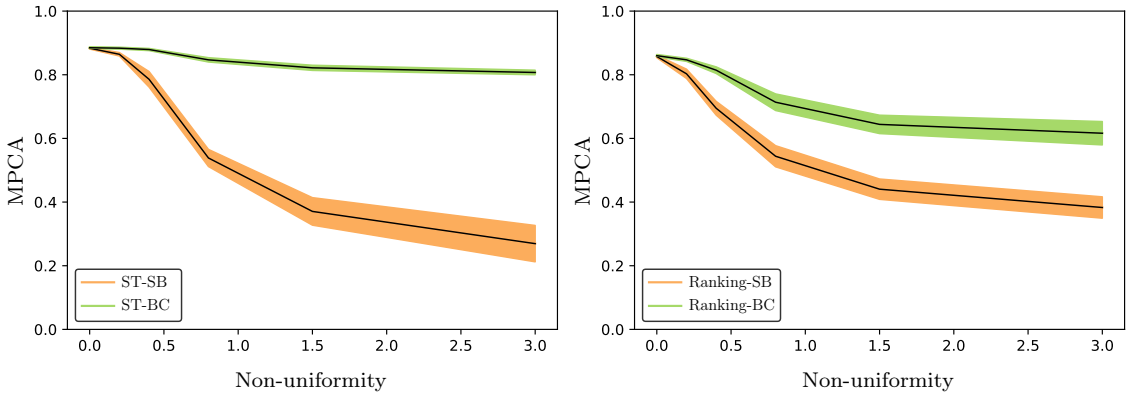


Figure 3.8: Mean per-class accuracy results on the MNIST dataset, for the levels of dataset non-uniformity given in Figure 3.6. Solid lines indicate the average over multiple runs, and the shaded regions indicate one standard deviation from that average. Left: standard batching vs batch construction with the cross-entropy loss. Right: standard batching vs batch construction with the ranking loss.

For the uniform case (with a non-uniformity level of 0), the MPCA is the same across all loss functions and batch selection strategies. This is expected, as batch construction and standard batching will yield similar training data distributions. There is a slight difference in that the number of times a class is sampled in a batch during standard batching is fixed. It would appear that this difference does not have a significant impact.

It is difficult to say whether the online ranking loss creates representations that are better for classification than the cross-entropy loss. Figure 3.9 shows results for settings 3c and 3d. The ranking loss outperforms the cross-entropy loss function in the standard batching case. So, while the ranking loss provides some robustness to a long-tailed distribution it appears that batch construction contributes more to such robustness. Additionally, while batch construction and the ranking loss both yield improved performance, using them in conjunction is not recommended. Training with the cross-entropy loss under batch construction seems to be better. This might be attributed to the fact that the pretrained network has already been trained with a cross-entropy classification loss and the ranking loss is not designed to optimise classification directly.

One way to think about batch construction and semi-hard negative mining (for the ranking loss) is that these strategies introduce implicit oversampling or undersampling, so that the distribution of seen training samples is such that more effective representations of data can be learned. Standard batching poses no prior on which samples are potentially better for learning. Under class-selective batch construction, however, the long-tailed

Stellenbosch University https://scholar.sun.ac.za

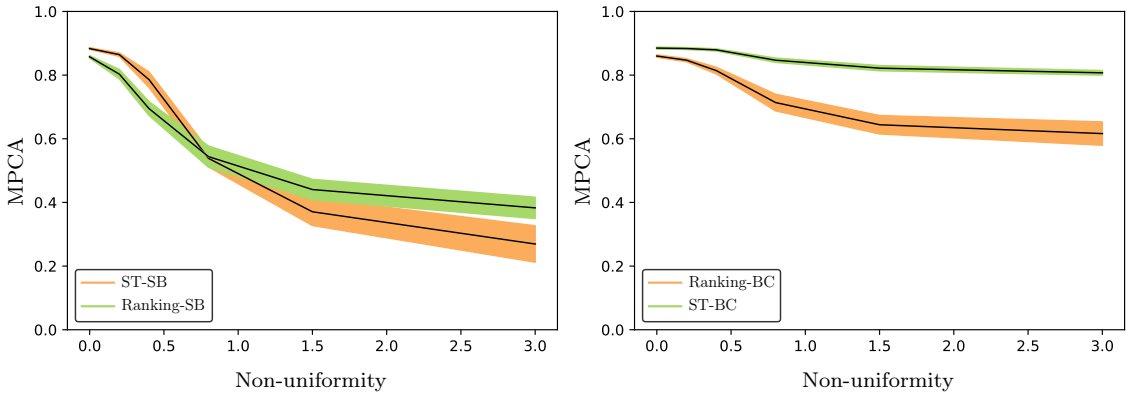


Figure 3.9: Mean per-class accuracy results on the MNIST dataset, for the levels of dataset non-uniformity given in Figure 3.6. Left: cross-entropy vs ranking loss with standard batching. Right: cross-entropy vs ranking loss with batch construction.

distribution becomes less significant since the dominant class is not always presented to the model during training.

CIFAR10

MNIST is a relatively simple dataset and we might be interested in whether the observed trends hold for more complex datasets. CIFAR10 is one such dataset where there are still 10 classes but they now represent natural images.

For settings 3a and 3b, trends are not so different from those observed on the MNIST dataset, as seen in Figure 3.10. Class-selective batch construction still offers improved performance in the longer tailed distributions. However, the improvement induced by batch construction is not as pronounced when coupled with the ranking loss. This raises the following question: under what learning conditions is batch construction effective? And, how much improvement can one expect when implementing batch construction? These questions, while interesting, will be deferred to future work. Recall that in these experiments there are two forms of training data distribution search occurring. Perhaps for more complicated datasets, batch construction does not offer any significant benefit to the implementation of a ranking loss with semi-hard negative mining. This idea will be further explored in Chapter 4, where it is tested on visual relationship recognition.

Figure 3.11 shows results for training settings 3c and 3d. As before, trends tend to be more or less the same as in the MNIST case, with the exception being that differences are either more, or less pronounced. With the complexity of the classification problem increased, it would appear that the ranking loss is even more robust in the extreme case. However, since the MPCA is relatively lower, the significance of the gap is questionable.

So, class-selective batch construction seems to be effective at improving performance on the tail of the distribution, despite its relative simplicity. Moreover, the ranking loss and data distribution search provide some benefit, but it is not clear how to predict the degree of benefit for new problems. We have thus seen that the choice of our models really shape the representations that are formed, by imposing certain priors. Neural networks

Stellenbosch University https://scholar.sun.ac.za

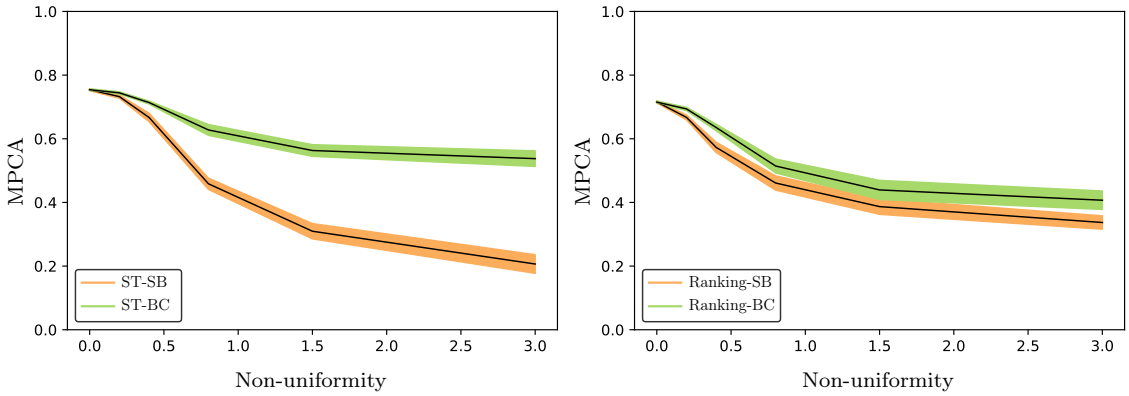


Figure 3.10: Mean per-class accuracy results on the CIFAR10 dataset, for the levels of dataset non-uniformity given in Figure 3.6. Left: standard batching vs batch construction with the cross-entropy loss. Right: standard batching vs batch construction with the ranking loss.

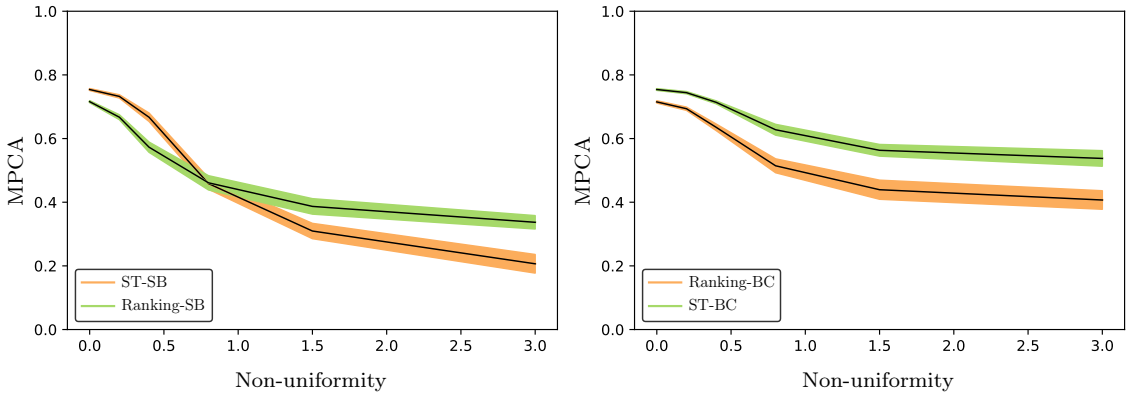


Figure 3.11: Mean per-class accuracy results on the CIFAR10 dataset, for the levels of dataset non-uniformity given in Figure 3.6. Left: cross-entropy vs ranking loss with standard batching. Right: cross-entropy vs ranking loss with batch construction.

create representations that exploit depth, abstraction, and a hierarchy of features that disentangle factors of variation. Since neural networks can be thought of as manipulating the data manifold, perhaps a ranking loss function might lead to some level of robustness to both a long-tailed distribution and fewer training samples; challenges inherent to problems such as visual relationship recognition. On the other hand, perhaps training data distribution search through batch construction is more beneficial. The next chapter explores these ideas further in the context of visual relationship recognition.

Stellenbosch University https://scholar.sun.ac.za

Chapter 4

Experiments and discussion

The previous chapter experimented with class-selective batch construction and data distribution search via semi-hard negative sampling in the ranking loss, in what can be called toy problems. This chapter explores multitask learning, in addition to the aforementioned techniques in the context of visual relationship recognition. We start by briefly reminding the reader of the problem description and providing summaries of our various models. Experimental analysis is thereafter broken into two stages. The first is on predicting the individual elements of a visual relationship triplet. The second is on the prediction of the entire visual relationship triplet. Both quantitative and qualitative evaluations are performed on the task of visual relationship recognition, to provide some insight into the behaviour of the various models. Results in all the tables and all the graphs in this chapter are given in percentages. For brevity, we refer to class-selective batch construction as "batch construction".

4.1 Summary of data models

Table 4.1 summarises all the versions of models that are studied in this chapter. Each model takes as input an image cropped on the union bounding box around a pair of objects, resized to $224 \times 224 \times 3$. Take note that the ranking loss with semi-hard negative mining is used in the hierarchical multitask and split-multitask training setups. The models are all evaluated on the VRD dataset, as introduced in Chapter 3. Recall that VRD contains 100 subject, 70 predicate, and 100 object labels that result in 700,000 possible visual relationships.

All models are implemented in the PyTorch framework [44]. For standard batching we use a batch size of 300. For class-selective batch construction we choose $n = 50$ (the number of classes to select per mini-batch) and $m = 6$ (the number of instances to sample per selected class). There could a trade-off in performance between the number of classes and sampled instances per class, but informal experimentation showed no significant difference for our models (which might be somewhat surprising, although it could be that effects average out over multiple batches). Mini-batch gradient descent is performed using Adam [45] with a learning rate scheduler that decreases the learning rate every eight training iterations. The parameter that controls this decrease was left as the default value. All training is performed on a single NVIDIA GeForce RTX 2070.

Stellenbosch University https://scholar.sun.ac.za

Table 4.1: Keys for the different models, grouped by training setup. Each block contains a model under a standard batching strategy followed by three batch construction strategies. The final eight models are trained with the ranking loss, using semi-hard negative mining as a form of training data distribution search.

Model	Description
ST-SB	single-task, standard batching
ST-BC-S	single-task, batch construction from subject labels
ST-BC-P	single-task, batch construction from predicate labels
ST-BC-O	single-task, batch construction from object labels
BMT-SB	basic multitask, standard batching
BMT-BC-S	basic multitask, batch construction from subject labels
BMT-BC-P	basic multitask, batch construction from predicate labels
BMT-BC-O	basic multitask, batch construction from object labels
HMT-SB	hierarchical multitask with ranking loss, standard batching
HMT-BC-S	hierarchical multitask with ranking loss, batch construction from subject labels
HMT-BC-P	hierarchical multitask with ranking loss, batch construction from predicate labels
HMT-BC-O	hierarchical multitask with ranking loss, batch construction from object labels
SMT-SB	split-multitask with ranking loss, standard batching
SMT-BC-S	split-multitask with ranking loss, batch construction from subject labels
SMT-BC-P	split-multitask with ranking loss, batch construction from predicate labels
SMT-BC-O	split-multitask with ranking loss, batch construction from object labels

4.2 Predicting individual elements

Results from the various models are presented in Table 4.2, for the three tasks of predicting the subject, the predicate and the object over all the samples in the test set. Note that although the results reported in this chapter are obtained from a single run of the models and can fluctuate due to randomness in the training procedure, informal experimentation showed fairly stable results across multiple runs. Performance according to the MPCA metric and R@1 metric is summarised in Figure 4.1 and Figure 4.2 respectively. In these figures, batch construction refers to class-selective batch construction with respect to the particular label on the horizontal axis.

Based on the MPCA metric we may note that batch construction offers a significant improvement in performance on the long tail-end of each individual task, but only if batches are constructed according to those same tasks. Figure 4.1 demonstrates this in that models with batch construction (the orange bars) perform consistently better across all models than their standard batching counterparts. Batch construction ensures a uniform label distribution for a particular task and as a result, MPCA is improved. On the contrary, batch construction based on another task seems to decrement MPCA for the subject and object tasks, except for the models where the ranking loss is used. It is not clear how batch construction based on labels from one task influences the distribution of other tasks and so explaining the decrement becomes difficult. By introducing the online ranking loss with semi-hard negative mining, however, this decrement is mitigated. It may be that data distribution search in the form of semi-hard negative mining promotes a training data distribution that is marginally closer to a more desired data distribution. While the inner-workings of data distribution search is not fully explored in this thesis, there is evidence that it can allow for better modelling of the tail of the distribution.

Stellenbosch University https://scholar.sun.ac.za

Table 4.2: Quantitative test results from various the various models, on predicting single elements of visual relationships.

Model	Subject prediction			Predicate prediction			Object prediction		
	MPCA	R@1	R@3	MPCA	R@1	R@3	MPCA	R@1	R@3
ST-SB	19.09	53.29	73.63	4.51	31.96	56.77	28.92	**40.23**	68.17
ST-BC-S	**33.13**	16.14	39.39	4.13	19.26	41.36	22.44	38.20	63.74
ST-BC-P	16.70	49.55	68.86	17.01	10.78	31.72	25.20	34.99	61.50
ST-BC-O	16.67	52.66	71.43	5.24	27.93	51.39	**40.72**	26.62	50.58
BMT-SB	19.96	**53.44**	**74.62**	4.74	**32.24**	**57.12**	28.34	40.03	**68.94**
BMT-BC-S	32.83	17.37	43.41	4.09	19.35	40.70	22.46	38.46	64.92
BMT-BC-P	17.18	50.26	70.95	**17.54**	12.71	32.39	26.24	35.59	62.65
BMT-BC-O	17.52	53.05	72.08	6.27	28.34	52.06	40.60	27.33	51.91
HMT-SB	5.96	48.57	62.96	3.42	30.00	52.71	6.70	29.83	51.68
HMT-BC-S	20.70	10.41	24.82	2.34	17.76	35.21	13.98	32.97	56.22
HMT-BC-P	11.09	46.50	62.68	14.88	7.80	23.94	12.70	29.10	49.86
HMT-BC-O	10.54	49.13	65.29	3.76	26.51	47.85	23.81	12.93	30.53
SMT-SB	10.08	49.91	66.94	3.74	30.45	54.31	14.83	34.49	59.11
SMT-BC-S	29.53	15.05	35.14	3.31	19.39	40.51	19.90	36.01	62.38
SMT-BC-P	16.68	50.12	70.21	15.57	13.19	36.03	23.23	34.57	61.34
SMT-BC-O	15.35	51.88	69.80	4.94	27.55	50.08	34.91	22.87	44.22

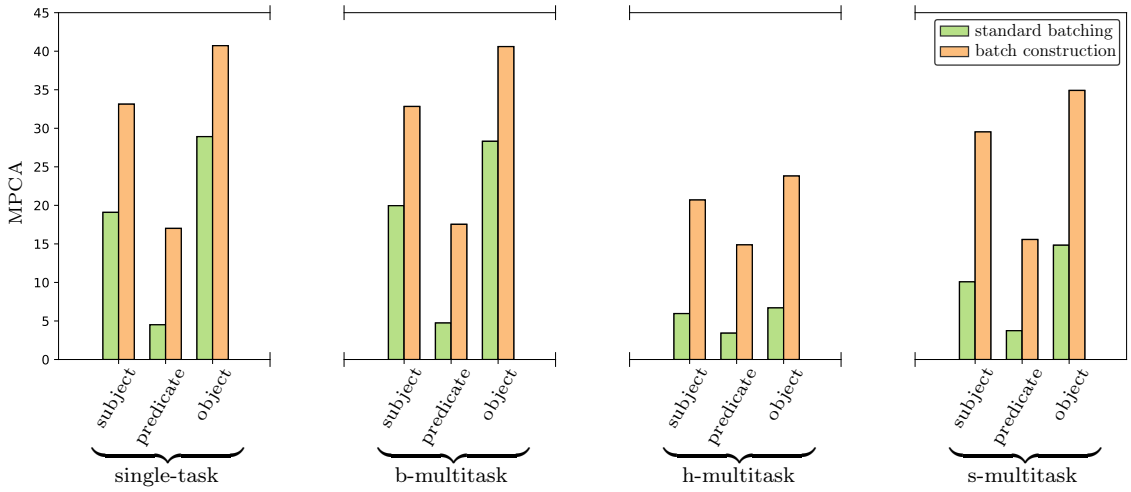


Figure 4.1: MPCA in predicting single relationship elements, with standard batching vs batch construction strategies. Bars that are connected can be compared directly. Higher bars indicate better performance on the tail of the distribution. The graph only shows results for batch construction with respect to the particular label on the horizontal axis (the same element that is being predicted).

Relatively lower accuracies from all models for the prediction of predicates verify the suspicion that predicates are harder to recognise visually, and might also be due to the greater diversity in the visual representation of predicates. The interaction between objects is also in a sense abstract and so models pretrained on object classification datasets might prevent effective transfer learning for predicates. Models could be trained from scratch to test this hypothesis, but this will require a dataset larger than VRD. There is a contrasting relationship between subject and object scores in Figure 4.1 and Figure 4.2. This relationship is interesting since in VRD, subjects and objects share the same label

Stellenbosch University https://scholar.sun.ac.za

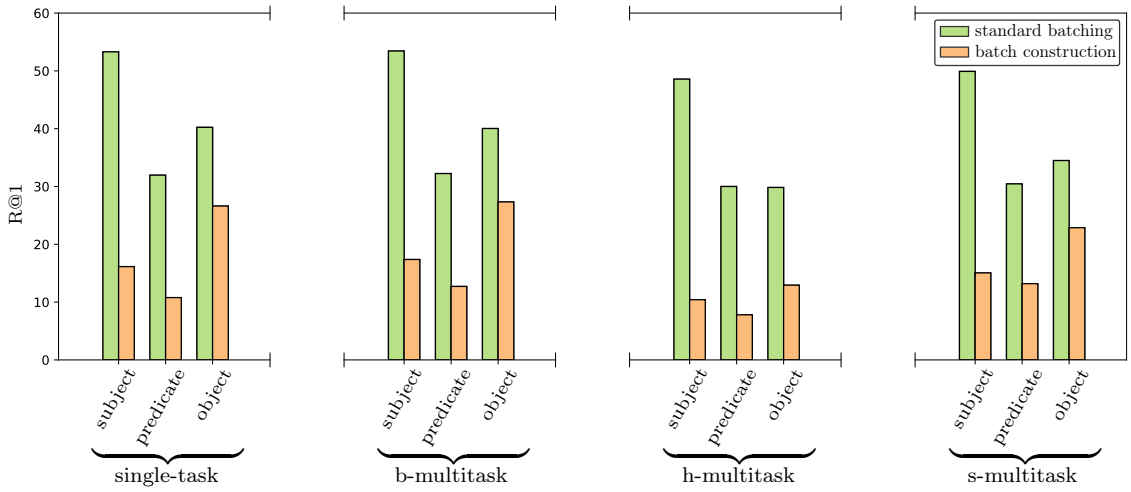


Figure 4.2: R@1 in predicting single relationship elements, with standard batching vs batch construction strategies. Bars that are connected can be compared directly. Higher bars indicate better performance on the task of classification overall. The graph only shows results for batch construction with respect to the particular label on the horizontal axis (the same element that is being predicted).

set. For MPCA, object scores are always higher than subject scores in both batching strategies. For R@1, the converse holds; subject scores are always higher than object scores. Higher subject scores for R@1 can be explained by the severity of the long tail and the fact that R@1 is biased towards the dominant classes. A model just needs to perform well on dominant classes to obtain a high R@1 score. Lower subject scores for MPCA might indicate that batch construction needs to be tailored to the severity of the long-tailed distribution.

The results in Table 4.2 also indicate higher R@1 and R@3 scores for models trained with standard batching compared to those that implement batch construction. This is more clearly depicted in Figure 4.2 where the bars corresponding to standard batching are consistently higher. There seems to be a trade-off: batch construction contributes to better generalisation on the many rare classes, at the cost of the accuracy on a small number of dominant classes. Moreover, batch construction with respect to the object labels deals with this trade-off better than in other tasks, as seen in the smaller difference between standard batching and batch construction in Figure 4.2. This could be due to the fact that objects are easier to recognise than predicates, and suffer from a far less severe long-tailed distribution than subjects. The severity of the long-tailed distribution of the subject labels induces interesting behaviour that will be discussed in Section 4.3.2.

Furthermore, we note that basic multitask learning does not seem to significantly improve or worsen mean per-class accuracy on the prediction of individual elements. Generally speaking, it is not yet clear under which circumstances a multitask model will improve performance, but there are arguments suggesting that more uniform label distributions in the auxiliary tasks might be preferred for multitask learning to be effective [46]. In our case the standard multitask models do provide similar performance to the multiple single-task models, which is useful if there are limitations on model size and complexity.

Stellenbosch University https://scholar.sun.ac.za

Reduced model capacity can also act as a form of regularisation.

When extending the multitask setup by introducing the ranking loss as an additional objective, both MPCA and R@1 are reduced. Jointly learning a semantic embedding space and classification scores in this context does not offer improved performance. The use of the ranking loss introduces strong regularisation, however, even when dropout is removed. We will later examine differences in behaviour of models trained with and without the ranking loss. It is worth reiterating that the ranking loss with semi-hard negative mining combats the decrement in MPCA in the subject and object tasks when performing batch construction with respect to a different task label.

A fairer comparison to be made would be between hierarchical multitask and split-multitask learning. Jointly training a shared representation with the ranking loss and a classifier for each task offers improvement in both R@1 and MPCA. It might be that hierarchical multitask learning regularises too strongly, as the entire network is shared. In the context of multitask learning, one would say that the training signals from the ranking loss destructively interfere with the classifier's gradient updates. The reduced accuracy could also be explained by hardware limitations. When using the ranking loss for face verification [34], Schroff et al. used batches of size 1,800, which is 6 times the size of batches used in our experiments. Perhaps increasing the search space for semi-hard negatives would allow for a more optimal choice of training samples. Informal experimentation showed that training the ranking loss models in a two-stage process, as in Section 3.3, yields inferior results and demonstrated that end-to-end training is better.

4.3 Predicting visual relationship triplets

4.3.1 Quantitative results

Table 4.3 shows evaluation results from the different models predicting full (subject, predicate, object) triplets. The models are exactly the same as those in the previous section, and still classify the three elements of a triplet separately, but we now evaluate their ability to predict all three elements correctly for a given input image crop. We report R@50 and R@100, as is standard in the literature, and remind the reader that there are 700,000 possible classes in this case, and a random classifier would get an R@100 of about 0.026% on our test set. Obtained results are quite similar to related work, but since we focus only on the labelling of visual relationships, and not also the localisation of individual objects, a direct comparison would not mean much. We will, however, test sensitivity to the tight bounding box assumption at the end of this chapter.

As before, standard batching produces better recall-at-50 and recall-at-100 compared to batch construction. However, when considering performance only on the long tail of the distribution (as explained at the end of Section 3.2), we find that batch construction does offer an improvement. Figure 4.3 also demonstrates this trade-off. For brevity, the graphs in this figure only compare standard batching and batch construction with respect to the object labels. The strikingly low values for hierarchical multitask and split-multitask learning suggest that the ranking loss, with a limited batch size, is unsuitable for the problem of visual relationship recognition. Batch construction seems to be a

Stellenbosch University https://scholar.sun.ac.za

Table 4.3: Quantitative test results from the various models, on predicting full visual relationship triplets.

Model	Predicting the full triplet			
	R@50	R@100	Tail R@50	Tail R@100
ST-SB	49.18	58.18	13.10	17.74
ST-BC-S	23.87	30.84	**20.96**	**27.82**
ST-BC-P	31.79	42.10	16.93	23.58
ST-BC-O	40.66	48.58	18.95	24.59
BMT-SB	**50.27**	**59.69**	12.50	18.95
BMT-BC-S	24.95	32.37	19.35	27.21
BMT-BC-P	33.56	44.08	17.94	26.41
BMT-BC-O	41.83	49.47	20.76	26.20
HMT-SB	36.15	42.87	0.40	0.60
HMT-BC-S	17.16	22.53	4.03	6.85
HMT-BC-P	21.19	29.98	7.25	11.08
HMT-BC-O	29.37	36.23	4.44	7.29
SMT-SB	41.23	48.79	2.42	5.04
SMT-BC-S	22.57	29.61	13.51	20.36
SMT-BC-P	34.85	44.61	17.94	24.19
SMT-BC-O	37.48	45.46	14.11	19.96

more effective means of data distribution search than semi-hard negative mining, as the Siamese networks seem to favour the dominant classes when standard batching is performed. Once again, perhaps the small batches are not appropriately representative for semi-hard negative mining to be effective.

While the ranking loss might add useful regularisation, there is too much of a performance detriment to warrant application. Having said that, the fact that split-multitask learning shows an improvement over hierarchical multitask learning suggests that there might be more effective approaches involving the ranking loss that were not considered in this thesis.

One may postulate that the predicate is most representative of the visual relationship, since the predicate defines the interaction between objects, but it appears from our experiments that batch construction with the object labels is a better strategy. The ResNet-18 layers might have an influence here, since they have been pretrained for object classification and are thus potentially less suited for the more abstract concepts of predicates. With that in mind, one may expect to gain similar benefit from subject-based batch construction, but as before, the severity of the long tail hinders learning.

By keeping the batching strategy constant we see in Figure 4.4 that single-task learning and basic multitask learning offer the best performance. A fundamental difference between the single- and multitask settings is that the latter receives a training signal from every element of the visual relationship triplet simultaneously. In some sense it sees the full visual relationship, yet the basic multitask setting does not yield significantly better scores compared to the single-task setting. Again, this shows that it is not immediately obvious that multitask learning will lead to improved metrics, and corroborates previous findings [46].

Stellenbosch University https://scholar.sun.ac.za

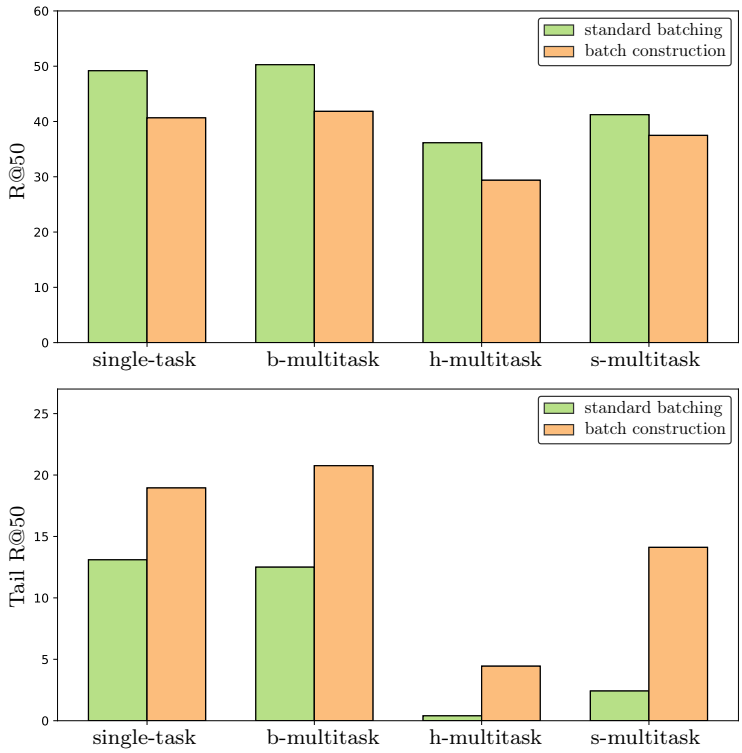


Figure 4.3: Top: R@50 for predicting the full relationship triplet on the test set, showing standard batching vs batch construction. Bottom: R@50 on the tail end of the test data. Note that scales on the y-axes are different.

4.3.2 Behaviour analysis

In the previous experiments we were interested in whether the correct (subject, predicate, object) triplet occurs in the top 50 (or 100) predicted triplets. If this is not the case, there are a few specific outcomes that may still be of interest. To gain deeper insights into the behaviour of the models we consider five mutually exclusive events, each predicated on all preceding events not taking place. The list below describes each event. We colour the predicate element according to whether it is correct (green) or incorrect (red).

1. **subject**, **predicate**, **object**: The correct visual relationship triplet occurs in the top 50 predictions. This is what the R@50 metric picks up when determining its score.

2. **subject**, **predicate**, **object**: Event 1 does not occur, but the correct subject and object appear together in the top 50 predictions with an incorrect predicate.

3. **object**, **predicate**, **subject**: Events 1 and 2 do not occur, but the three correct elements appear together in the top 50 just with the subject and object swapped.

4. **object**, **predicate**, **subject**: Events 1, 2 and 3 do not occur, but the correct subject and object appear together in the top 50, swapped and with an incorrect predicate.

Stellenbosch University https://scholar.sun.ac.za

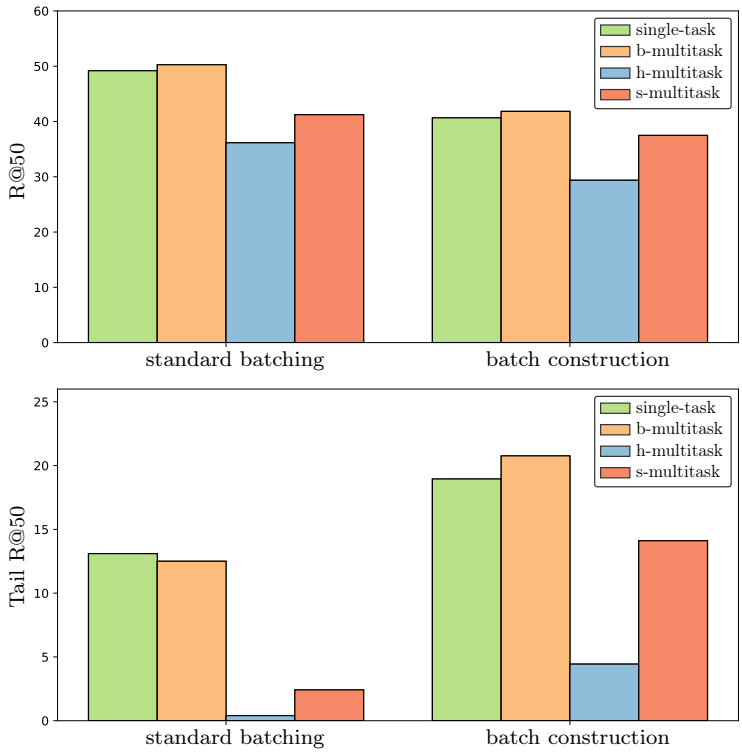


Figure 4.4: Top: R@50 for predicting the full relationship triplet on the test set, from various different models. Bottom: R@50 on the tail end of the test data. Note that scales on the y-axes are different.

5. **other**: Events 1, 2, 3 and 4 do not occur, that is, the correct subject and object do not appear together (in order or swapped) in the top 50 predictions.

Figure 4.5 shows the percentages of these events, for all the different models, across the test set. Note that this figure also offers a means of comparing R@50 scores across the models and quite clearly shows that object-based batch construction outperforms subject- and predicate-based batch construction. Later evaluations in this chapter will be limited to comparisons between standard batching and object-based batch construction.

Despite significant differences in implementation, all models find the correct subject and object with an incorrect predicate (event 2) at a similar rate. This once again suggests that the task of predicting the predicate is more challenging than that of the subject and object, even when batch construction ensures a uniform label distribution for the predicates. This analysis is consistent even when the subject and object labels are confused (event 4): the correct predicate is not found in the top 50 predictions at a similar rate when compared to standard batching. Perhaps there is not enough visually discriminative information in a predicate for a vision based classifier to be effective. Considering the many ambiguous predicate labels in the VRD dataset (refer to Figure 3.3), it is also likely that incorrect but semantically sensible predicates are being predicted. Quantitatively evaluating semantic similarity would require significant manual labour or clever use of a language model. The latter could make for a fruitful future research direction.

Stellenbosch University https://scholar.sun.ac.za

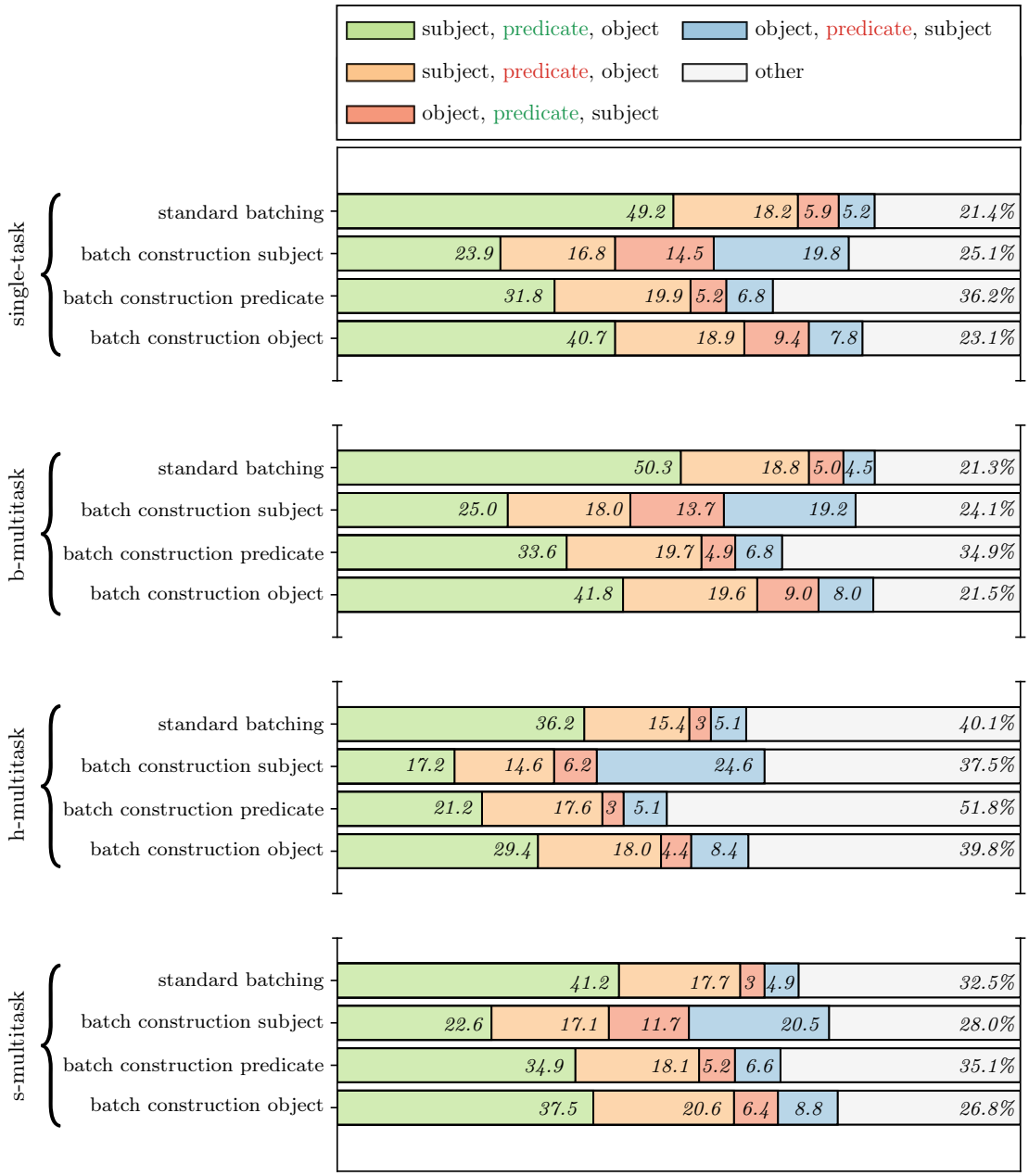


Figure 4.5: Behaviour analysis for the five events described in the text. Green and red in the legend are used to indicate whether the predicate is correctly or incorrectly classified.

Subject-based batch construction seems to consistently confuse the object and subject labels at a greater rate than all other models. This might be attributed to the severity of the long tail in the distribution over subject labels. The dominant subject class has around 15,000 samples, compared to the dominant object and predicate labels that have around 6,000 each. Performing subject-based batch construction lessens the strong bias that exists in the subject labels, and perhaps as a result the models tend to confuse the subject and object assignments. On the other hand, it should be noted that when a model confuses the subject and object, then missing the correct predicate (event 4)

Stellenbosch University https://scholar.sun.ac.za

occurs more frequently than picking up the correct predicate (event 3). This might be an indication that models swap the subject and object labels and reverse the predicate. For example, it is possible to have (`giraffe`, `taller than`, `person`) as the ground truth but have (`person`, `shorter than`, `giraffe`) as an acceptable answer that would be regarded as incorrect by our quantitative evaluations. A limitation of this evaluation is that difficult to say precisely what proportion of outcomes that are categorised by the given events should be classified as a correct prediction. To determine this, we would need to manually check the model predictions but that is tedious and time consuming, and quickly becomes infeasible for larger datasets. Alternatively, we could consider automating the process, by constructing a mapping of semantically similar predicates for event 2, predicates with a reflective property for event 3 and a mapping of inverse predicates for event 4. Since the number of labels in each respective task is relatively low, such a strategy might be feasible.

This kind of behaviour analysis, although to our knowledge not used in the literature, is useful since it shows the rate at which specific kinds of misclassifications are made. These kinds of misclassifications could potentially be semantically correct, and our evaluations motivate that semantics are important in visual relationship recognition. Specifically, dealing with semantic ambiguities may significantly improve results. For example, events 2, 3, and 4 might be equivalent to event 1 and would lead to a more exact evaluation. Moreover, it appears that categorically labelled data is incapable of capturing the necessary semantics to be effective in a discriminative sense. A language model might be of great use here, and will be discussed in the future work section of this thesis.

In light of the ambiguities that exist in visual relationships, comparing model predictions to a particular ground truth label does not offer the complete picture. To further investigate the behaviour of our models, we consider their outputs qualitatively in the next section.

4.3.3 Qualitative evaluation

Figures 4.6 and 4.7 show a number of test image samples and the top five predicted triplets from eight of the models. Figure 4.6 shows randomly chosen examples where the ST-SB model gives the correct visual relationship in its top five predictions, whereas Figure 4.7 shows randomly chosen examples where the ST-SB model does not give the correct visual relationship in its top five predictions. We choose to highlight batch construction based only on the object labels, since it performed best overall in the quantitative evaluations.

For the second example shown in Figure 4.6, (`giraffe`, `taller than`, `giraffe`), ST-BC-O gets the subject and object right but confidently predicts `in front of` as the predicate; perhaps a forgivable error. Similarly sensible errors can be seen throughout the examples in Figure 4.6, and demonstrate the ambiguities present in visual relationships. It is interesting to note that the `behind` and `in front of` predicates, although appearing in the top five predictions, have very different confidence scores. This is an example of an undesirable behaviour that a human would not make. It further motivates the inclusion of multi-modal semantics in the modelling process.

We also note that `person` is the dominating subject class, and is predicted correctly in almost all cases shown. Some predicted relationships for the (`person`, `on`, `horse`) example

Stellenbosch University https://scholar.sun.ac.za

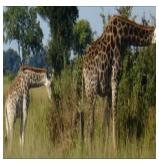
Models	person, on, horse 		giraffe, taller than, giraffe 		car, behind, car 	
ST-SB	person, on, horse	12.0	giraffe, taller than, giraffe	25.1	car, behind, car	20.6
	person, ride, horse	7.0	giraffe, in front of, giraffe	20.8	car, in front of, car	13.1
	person, wear, horse	5.3	giraffe, next to, giraffe	9.5	car, next to, car	7.2
	person, has, horse	5.2	giraffe, above, giraffe	7.6	car, on, car	4.7
	person, on, person	3.1	giraffe, behind, giraffe	7.2	car, near, car	4.3
ST-BC-O	person, on, horse	18.7	giraffe, in front of, giraffe	98.6	car, next to, car	8.5
	person, has, horse	11.8	giraffe, taller than, giraffe	0.4	car, behind, car	7.8
	person, wear, horse	7.7	giraffe, behind, giraffe	0.4	car, in front of, car	5.0
	person, in front of, horse	4.3	giraffe, next to, giraffe	0.1	car, next to, van	3.8
	person, next to, person	3.7	giraffe, beside, giraffe	0.1	car, has, car	3.7
BMT-SB	person, wear, horse	9.3	giraffe, taller than, giraffe	45.4	car, behind, car	14.1
	person, on, horse	6.8	giraffe, in front of, giraffe	18.9	car, in front of, car	11.6
	person, wear, person	3.4	giraffe, next to, giraffe	8.6	car, next to, car	7.2
	person, behind, horse	3.1	giraffe, behind, giraffe	7.3	car, on, car	4.6
	person, has, horse	2.6	giraffe, under, giraffe	2.6	car, near, car	3.4
BMT-BC-O	person, on, horse	13.2	giraffe, in front of, giraffe	92.5	car, behind, car	8.7
	person, above, horse	12.0	giraffe, taller than, giraffe	6.0	car, in front of, car	6.9
	person, behind, horse	6.3	giraffe, behind, giraffe	0.9	car, behind, van	5.3
	person, ride, horse	5.3	giraffe, next to, giraffe	0.3	car, in front of, van	4.3
	person, has, horse	4.8	giraffe, beside, giraffe	0.07	car, on, car	4.2
HMT-SB	person, on, motorcycle	3.0	sky, above, building	2.0	person, behind, bus	0.9
	person, on, person	2.0	sky, in front of, building	1.9	person, next to, bus	0.8
	person, behind, motorcycle	1.9	sky, behind, building	1.5	bus, behind, bus	0.7
	person, next to, motorcycle	1.8	building, above, building	1.4	bus, next to, bus	0.6
	person, has, motorcycle	1.8	building, in front of, building	1.3	person, on, bus	0.6
HMT-BC-O	person, on, bike	3.1	giraffe, taller than, giraffe	31.5	car, on, wheel	0.4
	person, on, horse	1.9	giraffe, taller than, bear	9.1	car, on, truck	0.4
	person, has, bike	1.9	trees, taller than, giraffe	7.7	car, behind, wheel	0.4
	person, on, motorcycle	1.7	giraffe, behind, giraffe	5.9	car, behind, truck	0.4
	person, on, basket	1.3	tree, taller than, giraffe	3.6	car, on, car	0.4
SMT-SB	person, wear, person	4.7	giraffe, taller than, giraffe	6.4	car, behind, car	4.9
	person, on, person	3.9	giraffe, behind, giraffe	4.9	car, on, car	4.6
	person, has, person	2.1	giraffe, in front of, giraffe	3.7	car, has, car	4.1
	person, behind, person	1.9	giraffe, above, giraffe	3.0	car, next to, car	2.9
	person, next to, person	1.7	giraffe, next to, giraffe	2.8	car, in front of, car	2.8
SMT-BC-O	person, on, horse	16.2	giraffe, in front of, giraffe	61.8	car, behind, car	4.2
	person, ride, horse	7.6	giraffe, taller than, giraffe	13.4	car, on, car	3.6
	person, behind, horse	4.7	giraffe, behind, giraffe	9.2	car, next to, car	3.4
	person, wear, horse	4.1	giraffe, next to, giraffe	5.4	car, in front of, car	2.8
	person, hold, horse	3.9	giraffe, under, giraffe	1.2	car, has, car	2.0

Figure 4.6: Top five visual relationship triplet predictions on three test images, where the models performed relatively well.

may seem nonsensical, like (`person`, `behind`, `horse`) or (`person`, `behind`, `person`). There is actually another person in the background of the image, and the models might be recognising the `person` class despite the minimal visual cues that exist.

Models trained with batch construction appear to make predictions with relatively high confidence scores. Recall that there are a total of 700,000 normalised confidence scores, so high scores in the top five predictions mean exceptionally low scores for the remaining 699,995 relationships. It is interesting that under an arguably more uniform training

Stellenbosch University https://scholar.sun.ac.za

data distribution, the confidence scores are this heavily skewed. Another observation regarding the confidence scores is that scores for models involving the ranking loss are relatively lower. Among those, HMT-BC-O shows relatively higher confidence scores. Predictions from the ranking loss models also vary more than those from models with the cross-entropy loss. In both figures, these varied predictions are often semantically similar to the ground truth. For example, `bear` is predicted instead of `giraffe`, and `giraffe` is predicted instead of `elephant`. This may be attributed to the fact that these models are less confident about predictions overall, or they are learning some kind of embedding in which semantically similar objects are mapped close to one another. The mechanism behind it is unclear at this stage, but the models may be learning the embedding through visual similarity or similar contexts across the training data. It may also suggest that models trained with the ranking loss are effective in their objective of learning a visual semantic embedding space.

The HMT-SB model produces nonsensical outputs over multiple examples, and further confirms that hierarchical multitask learning with the ranking loss may not be as effective as other methods.

In Figure 4.7 we see some examples where the models swap the positions of the subject and object labels and misclassify the predicate, as investigated in section 4.3.2. This is seen in the example where three of the models predict (`tree`, `next to`, `bear`) instead of the ground truth (`bear`, `adjacent to`, `tree`). These misclassifications may again be forgivable, especially also since `adjacent to` is one of the more obscure predicates.

In the first example of Figure 4.7, models predict predicates other than `on`, despite it being the dominating predicate class. This may be due to the strong visual cues in favour of interactions between the person and their items of clothing, rather than the more obscure `skateboard`.

The predicates `feed` and `adjacent to` are rare tail-end predicates that are misclassified even under batch construction strategies. The visual representation of these relationships do seem to be in line with the predicted relationships, however. Perhaps with the inclusion of multi-modal semantics in the training procedure, these kind of errors can be mitigated.

4.4 Bounding box perturbation

All of our models assume knowledge of tight bounding boxes around pairs of related objects. To conclude our experiments, we test the sensitivity of a few models to perturbations on the bounding boxes, for an indication of how the models would respond in practice when combined with an automatic object detector.

To measure the degree of perturbation, we use what is known as intersection over union (IOU). For two bounding boxes B_1 and B_2, the IOU is defined as

$$\text{IOU} = \frac{\text{area}(B_1 \cap B_2)}{\text{area}(B_1 \cup B_2)}, \tag{4.1}$$

Stellenbosch University https://scholar.sun.ac.za

Models	person, on, skateboard 		bear, adjacent to, tree 		person, feed, elephant 	
ST-SB	person, wear, person	11.8	bear, next to, grass	3.6	person, above, street	4.3
	person, wear, shirt	10.5	bear, on, grass	3.3	person, on, street	4.1
	person, wear, skateboard	10.0	bear, next to, person	2.5	person, under, street	3.0
	person, wear, shoes	5.4	bear, next to, tree	2.3	sky, above, street	1.7
	person, wear, pants	4.4	bear, on, person	2.3	sky, on, street	1.6
ST-BC-O	person, wear, skateboard	25.6	tree, next to, bear	78.7	person, under, elephant	16.4
	person, on, skateboard	10.0	bear, next to, bear	11.2	person, in front of, elephant	16.0
	person, has, skateboard	9.6	tree, near, bear	2.7	person, above, elephant	10.0
	person, ride, skateboard	5.2	person, next to, bear	0.7	person, near, elephant	4.7
	person, wear, shoes	3.5	tree, right of, bear	0.6	person, behind, elephant	4.1
BMT-SB	person, wear, shirt	15.5	bear, next to, grass	4.3	person, on, street	4.7
	person, wear, person	9.6	bear, next to, bear	3.5	person, under, street	3.9
	person, wear, skateboard	6.9	bear, on, grass	3.1	person, above, street	3.4
	person, wear, shoes	6.1	bear, on, bear	2.6	person, on, person	2.4
	person, wear, pants	4.1	bear, behind, grass	2.5	person, under, person	1.9
BMT-BC-O	person, wear, skateboard	20.0	tree, next to, bear	87.7	person, in front of, elephant	7.4
	person, wear, shoes	14.0	bear, next to, bear	2.7	person, near, elephant	6.9
	person, wear, helmet	12.0	tree, behind, bear	0.8	person, under, elephant	5.1
	person, has, skateboard	3.8	tree, beside, bear	0.8	person, on, elephant	3.4
	person, wear, pants	3.7	tree, next to, grass	0.7	person, above, elephant	2.4
HMT-SB	person, wear, person	7.3	person, on, wheel	0.8	person, on, street	1.3
	person, wear, shirt	5.3	person, has, wheel	0.7	person, above, street	0.9
	person, wear, pants	4.0	person, on, car	0.5	person, on, person	0.8
	person, wear, jacket	3.0	person, has, car	0.5	person, under, street	0.7
	person, has, person	2.9	person, next to, wheel	0.5	person, above, person	0.6
HMT-BC-O	person, wear, ball	8.0	person, next to, bear	2.7	person, above, horse	1.8
	person, has, ball	5.0	bear, next to, bear	2.3	person, above, bench	1.4
	person, wear, skateboard	4.6	person, behind, bear	1.7	person, above, dog	0.8
	person, hold, ball	3.4	bear, behind, bear	1.6	person, above, grass	0.7
	person, has, skateboard	2.9	dog, next to, bear	0.9	person, behind, horse	0.7
SMT-SB	person, wear, person	6.7	bear, on, grass	1.0	person, above, street	1.6
	person, wear, pants	6.0	bear, on, bear	1.0	person, on, street	1.2
	person, wear, shirt	5.9	bear, next to, grass	1.0	person, above, person	1.1
	person, wear, jacket	3.7	bear, next to, bear	0.9	sky, above, street	1.0
	person, wear, hat	3.5	bear, on, street	0.7	person, under, street	1.0
SMT-BC-O	person, wear, shorts	5.1	tree, next to, bear	26.3	person, above, ramp	5.0
	person, wear, shoes	4.6	bear, next to, bear	12.2	person, above, giraffe	2.6
	person, wear, skateboard	4.2	tree, behind, bear	6.7	person, under, ramp	2.5
	person, wear, ball	3.8	grass, next to, bear	5.1	person, below, ramp	2.3
	person, wear, shoe	3.6	bush, next to, bear	4.2	person, above, elephant	1.9

Figure 4.7: Top five visual relationship triplet predictions on three test images, where the models performed relatively badly.

and measures the percentage overlap between the two bounding boxes. Figure 4.8 illustrates. IOU, also referred to as the Jaccard index, is a standard metric used to evaluate object detectors.

To perturb the bounding boxes in the VRD dataset (recall Figure 3.2 for example), we add varying levels of Gaussian noise to the four corners of a bounding box. More specifically, we fix the mean of a Gaussian distribution from which noise is sampled at zero and use the following set of standard deviations: $\{0, 5, 10, 15, 20, 45, 55, 65, 75\}$. These standard

Stellenbosch University https://scholar.sun.ac.za

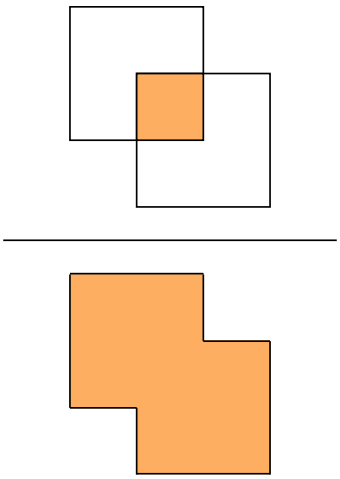


Figure 4.8: An illustration of intersection over union for two bounding boxes. The metric gives an indication of the overlap for two bounding boxes and is standard in object detection.

deviations result in intersection over union measures from 100% down to around 50%, which is the standard threshold used in the object detection literature.

When bounding boxes are perturbed, it is possible that the resulting box is outside the image entirely. If after 50 random perturbations any pixels of the resulting bounding box are not within image bounds, the ground truth bounding box for that sample is left as is. We show the resulting mean IOU between the original bounding boxes of the test set and their perturbations, as a percentage in Figure 4.9. It may also happen that perturbed bounding boxes remain within the image bounds but no longer contain the visual relationship at all, so we expect R@50 scores to decrease with increased noise levels.

Figure 4.9 shows results from the BMT-SB and BMT-SB-O models. Note that we use the same models tested in earlier sections, and do not retrain them on perturbed bounding boxes. As expected there is a decline in R@50 scores, but the decline occurs at a fairly low rate. Scores remain similar in multiple sections of the graph, which indicates some level of robustness under perturbations of the bounding boxes. The two different models seem to be affected in remarkably similar ways. The curves are almost exact replicas, with one just a constant offset away from the other. This may be due to errors averaging out across the test set, or it may be that there is an unmodelled factor at play which is not influenced by batch construction.

Stellenbosch University https://scholar.sun.ac.za

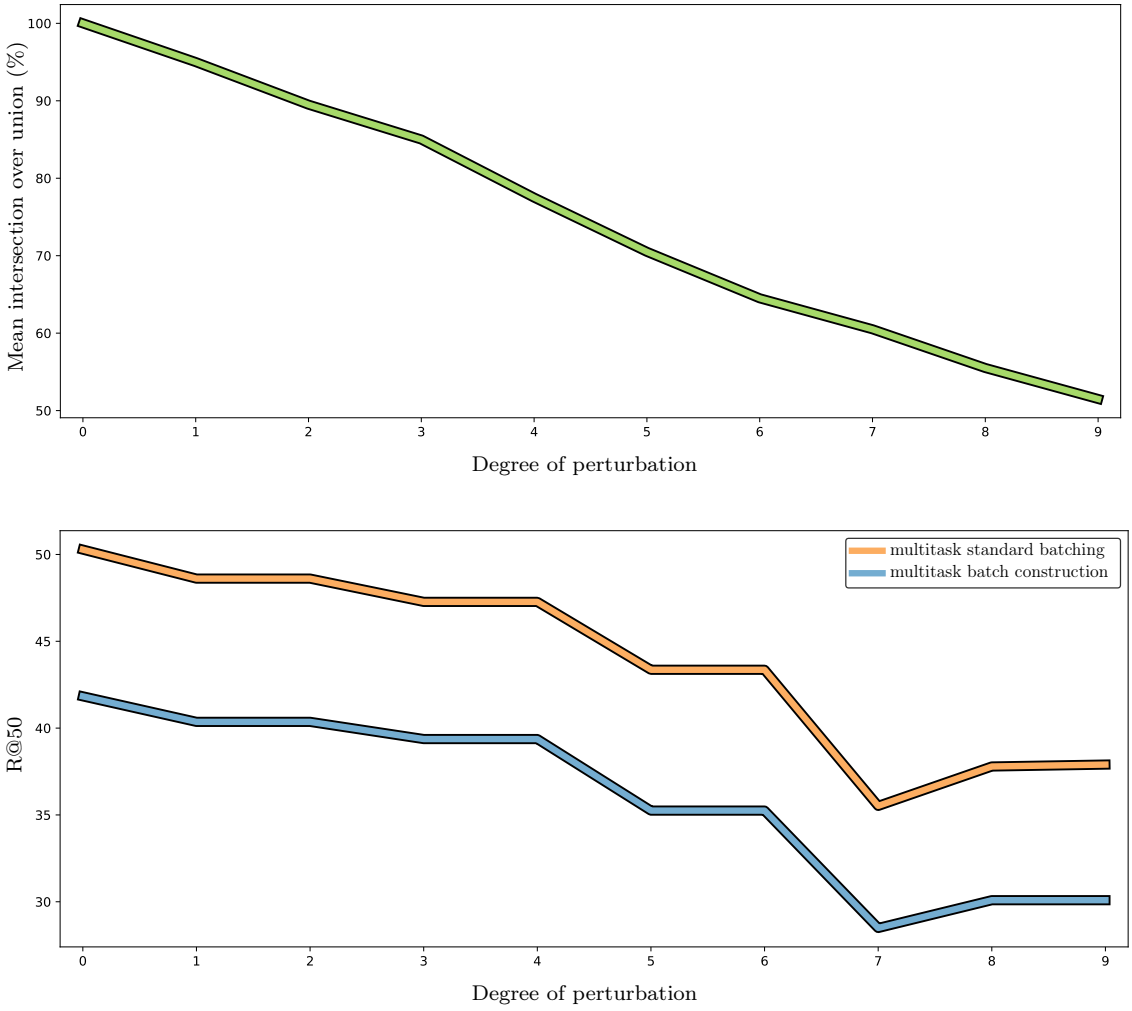


Figure 4.9: Top: intersection over union measure between the original bounding boxes and their perturbed versions, over varying levels of pertubation. The horizontal axis represents the indices of the set of standard deviations mentioned earlier. Bottom: R@50 on the perturbed test set for basic multitask with standard batching, and basic multitask with batch construction performed with respect to object labels.

Stellenbosch University https://scholar.sun.ac.za

Chapter 5

Conclusions and future work

This chapter offers a summary of findings and suggests a number of recommendations for future work. The recommendations are two-fold in the sense that they are concerned with modelling techniques and evaluation strategies. The latter is important since with appropriate feedback on model performance, research can iterate more efficiently.

The task in this thesis is visual relationship recognition. That is, given a bounding box around a pair of objects in an image, predict a visual relationship triplet of the form (subject, predicate, object).

5.1 Summary of findings

Visual relationship recognition is a challenging task in computer vision, given the large number of possible relationships as well as a typical long-tailed distribution over those relationships. We investigated the potential of class-selective batch construction and three variants of multitask learning. Two of the multitask variants involve the use of an online ranking loss function for the creation of an embedding space.

We saw that a class-selective batch construction strategy does improve performance on the tail of the distribution, but at the cost of performance on the small number of dominating classes at the head of the distribution. Batch construction only offers this improvement when performed with respect to the labels of the same task. That is, ensuring a uniform training data distribution within a batch, for a particular task, results in improvements on the tail. Additionally, there is some evidence that batch construction dampens the bias towards overly dominant classes within the distribution. Whether this behaviour is desirable is debatable.

We demonstrated that it is more difficult to model and recognise the predicate of a relationship, even when using predicate-based batch construction, and suggest that current pretrained models might not be particularly suitable for that task.

Basic multitask learning does not seem to improve or impede performance when compared to single-task learning, but provides other benefits such as a reduced model capacity. Reduced capacity is useful since training would require fewer resources and the risk of overfitting is lower. The variants of multitask learning that use the ranking loss regularise even stronger, but at the cost of performance across all metrics. Furthermore, split-multitask learning, where only part of the network is shared, offers improved performance over hierarchical multitask learning. Even though the ranking loss variants of multitask learning lead to lower performance, there are some indications that semantically similar objects are embedded closer together.

Stellenbosch University https://scholar.sun.ac.za

We offered a metric in addition to what is found in the literature, to analyse the frequency at which three specific types of errors occur in the top 50 predictions of a full relationship triplet. This metric can be viewed as an extension to the current recall-at-k metric, and provides deeper insight into how models behave. Further analysis on these types of errors can lead to significantly more effective models. We demonstrated these and a number of semantically sensible errors, through a few test examples.

Finally, we tested the sensitivity of a subset of our trained models to perturbations in the bounding boxes around pairs of interacting objects, and found some level of robustness to such perturbations.

5.2 Future work

5.2.1 Evaluation metrics

R@50 is limited in that it relies strongly on hard comparison with the ground truth. We have seen how misclassifications can often still be semantically correct due to ambiguity inherent in visual relationships. It may thus be useful to design additional metrics that can measure semantic similarity. WordNet [47] is a lexical database that contains what is known as synonym sets, or synsets for short. Nouns, verbs, adjectives or adverbs that express the same concept are grouped in a particular synset. Model outputs can then automatically be compared to synsets for better evaluation. The Visual Genome dataset [5] already contains mappings from class labels to WordNet synsets, but this information is not yet commonly used in model evaluation. A language model could also be used to measure semantics, in an effort to mitigate the inherent ambiguity in visual relationship labels.

5.2.2 Modelling semantics with language

A language model is a natural tool to use when attempting to model semantics. Lu et al. [4] employed a language model for the purpose of obtaining a visual relationship embedding space, but there may be other ways to do so.

The ranking loss can naturally be extended to include a language model. The distance function used in the ranking loss could include distances between word-vector embeddings as a notion of similarity. In addition to this, WordNet synsets can be used to indicate similarity between positive samples.

A language model can also be used to re-score the outputs of a classifier and thus encode semantics. In this way, model confidence in relationship triplets that would be unlikely from a semantic point of view, such as (`giraffe`, `drive on`, `umbrella`), can be suppressed. Techniques such as N-best list re-scoring [48; 49], and lattice re-scoring [50; 51] can be used. Language model re-scoring seems to be a common tool in automatic speech recognition systems.

Stellenbosch University https://scholar.sun.ac.za

5.2.3 Dynamic sampling

Class-selective batch construction and semi-hard negative mining are viewed as forms of data distribution search. They attempt to find ways to better select training samples so that models can generalise more effectively. The use of the ranking loss did not show improved results over the cross-entropy loss, so perhaps employing a type of data distribution search using cross-entropy can lead to improved performance.

Pouyanfar et al. [52] introduce the idea of dynamic sampling to deal with imbalanced datasets. They use the F1 scores on a validation set to tune the training set's class distribution and achieve improved results. Their method is an option for further exploration in visual relationship recognition. It is possible to extend the idea by using the cross-entropy loss values to select training samples, as one would for the ranking loss function. Figure 5.1 shows how this can be achieved in an online fashion. After computing the forward pass for a given batch, training samples are selected if the loss they incur is

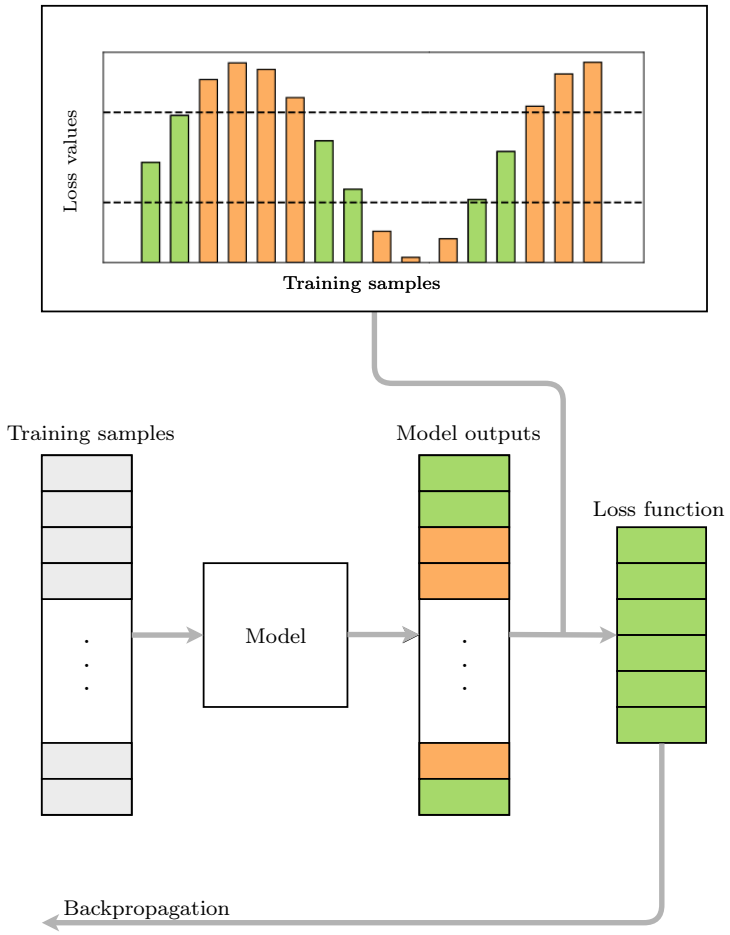


Figure 5.1: Dynamic sampling according to the cross-entropy loss function. Training samples whose loss values are within some range are coloured green, and only they are selected for computation of the loss and its gradient.

Stellenbosch University https://scholar.sun.ac.za

within some range. Only these samples are used to compute the loss function and its gradient, after which weights are updated.

There are a few questions that can be asked regarding these ideas, as well as batch construction:

- Is there a minimum batch size before training data distribution search becomes effective, especially under a long tail?
- Which range(s) of the cross-entropy will lead to a training data distribution that can improve generalisation, if any?
- Can training data distribution search be learned alongside the objective, as a form of meta-learning?

5.2.4 Scene graph generation

A subsequent task, after visual relationship recognition, is to generate the scene graph of an image, to be used for applications such as visual question answering, automated surveillance, and automatic monitoring of content on streaming platforms.

5.3 Concluding remarks

Visual relationship recognition is a challenging problem and the overarching theme is that there seems to be a trade-off between learning dominant classes and learning to recognise the many rare classes in the tail-end of the class distribution. We suggest further steps that encourage experimentation with new techniques as well as better model interpretation. Perhaps with some clever combination of these ideas, all parts of the data distribution can be effectively learned.

Stellenbosch University https://scholar.sun.ac.za

References

[1] Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.: You only look once: unified, real-time object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 779–788. 2016.

[2] Zhang, S., Wen, L., Bian, X., Lei, Z. and Li, S.: Single-shot refinement neural network for object detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4203–4212. 2018.

[3] Johnson, J., Krishna, R., Stark, M., Li, L.-J., Shamma, D., Bernstein, M. and Fei-Fei, L.: Image retrieval using scene graphs. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3668–3678. 2015.

[4] Lu, C., Krishna, R., Bernstein, M. and Fei-Fei, L.: Visual relationship detection with language priors. In: *European Conference on Computer Vision*, pp. 852–869. 2016.

[5] Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.-J., Shamma, D.A., Bernstein, M. and Fei-Fei, L.: Visual genome: connecting language and vision using crowdsourced dense image annotations. *arXiv preprint arXiv:1602.07332*, 2016.

[6] Caruana, R.: Multitask learning. *Machine Learning*, vol. 28, no. 1, pp. 41–75, 1997.

[7] Zhang, H., Kyaw, Z., Chang, S.-F. and Chua, T.-S.: Visual translation embedding network for visual relation detection. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5532–5540. 2017.

[8] Zhang, J., Kalantidis, Y., Rohrbach, M., Paluri, M., Elgammal, A. and Elhoseiny, M.: Large-scale visual relationship understanding. *arXiv preprint arXiv:1804.10660*, 2018.

[9] Xu, D., Zhu, Y., Choy, C.B. and Fei-Fei, L.: Scene graph generation by iterative message passing. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5410–5419. 2017.

[10] Zellers, R., Yatskar, M., Thomson, S. and Choi, Y.: Neural motifs: scene graph parsing with global context. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5831–5840. 2018.

[11] Newell, A. and Deng, J.: Pixels to graphs by associative embedding. In: *Advances in Neural Information Processing Systems*, pp. 2171–2180. 2017.

[12] Qi, S., Wang, W., Jia, B., Shen, J. and Zhu, S.-C.: Learning human-object interactions by graph parsing neural networks. In: *European Conference on Computer Vision*, pp. 401–417. 2018.

[13] Yang, J., Lu, J., Lee, S., Batra, D. and Parikh, D.: Graph R-CNN for scene graph generation. In: *European Conference on Computer Vision*, pp. 670–685. 2018.

[14] Woo, S., Kim, D., Cho, D. and Kweon, I.S.: LinkNet: relational embedding for scene graph. In: *Advances in Neural Information Processing Systems*, pp. 560–570. 2018.

Stellenbosch University https://scholar.sun.ac.za

[15] Dai, B., Zhang, Y. and Lin, D.: Detecting visual relationships with deep relational networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3076–3086. 2017.

[16] Wang, G., Luo, P., Lin, L. and Wang, X.: Learning object interactions and descriptions for semantic image segmentation. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5859–5867. 2017.

[17] Chao, Y.-W., Liu, Y., Liu, X., Zeng, H. and Deng, J.: Learning to detect human-object interactions. In: *IEEE Winter Conference on Applications of Computer Vision*, pp. 381–389. 2018.

[18] Yin, G., Sheng, L., Liu, B., Yu, N., Wang, X., Shao, J. and Change Loy, C.: Zoom-Net: mining deep feature interactions for visual relationship recognition. In: *European Conference on Computer Vision*, pp. 322–338. 2018.

[19] Gkioxari, G., Girshick, R., Dollár, P. and He, K.: Detecting and recognizing human-object interactions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8359–8367. 2018.

[20] Li, Y., Ouyang, W., Wang, X. and Tang, X.: ViP-CNN: visual phrase guided convolutional neural network. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1347–1356. 2017.

[21] Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T. and Ferrari, V.: The Open Images Dataset V4: unified image classification, object detection, and visual relationship detection at scale. *arXiv preprint arXiv:1811.00982*, 2018.

[22] Bengio, Y., Courville, A. and Vincent, P.: Representation learning: a review and new perspectives. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[23] Goodfellow, I., Bengio, Y. and Courville, A.: *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[24] Fefferman, C., Mitter, S. and Narayanan, H.: Testing the manifold hypothesis. *Journal of the American Mathematical Society*, vol. 29, no. 4, pp. 983–1049, 2016.

[25] Olah, C.: Neural networks, manifolds, and topology. 2014.
Available at: `https://colah.github.io/posts/2014-03-NN-Manifolds-Topology/`

[26] Hubel, D.H. and Wiesel, T.N.: Receptive fields of single neurones in the cat's striate cortex. *The Journal of Physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[27] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. and Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[28] Krizhevsky, A., Sutskever, I. and Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: *Advances in Neural Information Processing Systems*, pp. 1097–1105. 2012.

[29] He, K., Zhang, X., Ren, S. and Sun, J.: Deep residual learning for image recognition. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778. 2016.

[30] Simonyan, K. and Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Stellenbosch University https://scholar.sun.ac.za

[31] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A.: Going deeper with convolutions. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9. 2015.

[32] Huang, G., Liu, Z., Van Der Maaten, L. and Weinberger, K.Q.: Densely connected convolutional networks. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708. 2017.

[33] Olah, C., Mordvintsev, A. and Schubert, L.: Feature visualization. *Distill*, 2017. Available at: `https://distill.pub/2017/feature-visualization`

[34] Schroff, F., Kalenichenko, D. and Philbin, J.: Facenet: a unified embedding for face recognition and clustering. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823. 2015.

[35] Mitchell, T.M.: *The need for biases in learning generalizations.* Technical Report CBM-TR-117, Rutgers University, 1980.

[36] Ruder, S.: An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*, 2017.

[37] Wu, C.-Y., Manmatha, R., Smola, A.J. and Krahenbuhl, P.: Sampling matters in deep embedding learning. In: *IEEE International Conference on Computer Vision*, pp. 2840–2848. 2017.

[38] Bromley, J., Guyon, I., LeCun, Y., Säckinger, E. and Shah, R.: Signature verification using a "Siamese" time delay neural network. In: *Advances in Neural Information Processing Systems*, pp. 737–744. 1994.

[39] Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A. and Torr, P.H.: Fully-convolutional Siamese networks for object tracking. In: *European Conference on Computer Vision*, pp. 850–865. 2016.

[40] He, A., Luo, C., Tian, X. and Zeng, W.: A twofold Siamese network for real-time object tracking. In: *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4834–4843. 2018.

[41] Varior, R.R., Haloi, M. and Wang, G.: Gated Siamese convolutional neural network architecture for human re-identification. In: *European Conference on Computer Vision*, pp. 791–808. Springer, 2016.

[42] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P.: Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[43] Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, 2009.

[44] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S.: PyTorch: an imperative style, high-performance deep learning library. In: *Advances in Neural Information Processing Systems*, pp. 8024–8035. 2019.

[45] Kingma, D.P. and Ba, J.: Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[46] Alonso, H.M. and Plank, B.: When is multitask learning effective? Semantic sequence prediction under varying data conditions. *arXiv preprint arXiv:1612.02251*, 2016.

Stellenbosch University https://scholar.sun.ac.za

[47] Miller, G.A.: WordNet: a lexical database for English. *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[48] Stolcke, A., Konig, Y. and Weintraub, M.: Explicit word error minimization in n-best list rescoring. In: *European Conference on Speech Communication and Technology*, pp. 163–166. 1997.

[49] Verhasselt, J. and Dercks, H.: N-best list rescoring in speech recognition. *Acoustical Society of America Journal*, vol. 128, no. 6, p. 3828, 2010.

[50] Sundermeyer, M., Tüske, Z., Schlüter, R. and Ney, H.: Lattice decoding and rescoring with long-span neural network language models. In: *Fifteenth Annual Conference of the International Speech Communication Association*. 2014.

[51] Kumar, S., Nirschl, M., Holtmann-Rice, D., Liao, H., Suresh, A.T. and Yu, F.: Lattice rescoring strategies for long short-term memory language models in speech recognition. In: *IEEE Automatic Speech Recognition and Understanding Workshop*, pp. 165–172. 2017.

[52] Pouyanfar, S., Tao, Y., Mohan, A., Tian, H., Kaseb, A.S., Gauen, K., Dailey, R., Aghajanzadeh, S., Lu, Y.-H., Chen, S.-C. and Shyu, M.-L.: Dynamic sampling in convolutional neural networks for imbalanced data classification. In: *IEEE Conference on Multimedia Information Processing and Retrieval*, pp. 112–117. 2018.